Project Number: WND1

Design of an Autonomous Platform for Search and Rescue UAV Networks

A Major Qualifying Project Report
submitted to the faculty of WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the degree of Bachelor of Science

Submitted by:

Catherine Coleman

Robotics Engineering and Mechanical Engineering

Joseph Funk

Electrical and Computer Engineering

James Salvati

Robotics Engineering and Electrical and Computer Engineering

Christopher Whipple

Robotics Engineering

Advisors:

Professor Taskin Padir

Robotics Engineering and Electrical and Computer Engineering

Professor Alexander Wyglinski

Electrical and Computer Engineering

On

April 26th 2012

Abstract

In this report, we present a platform for use in a system of unmanned aerial vehicles (UAVs) capable of human assisted-autonomous and fully autonomous flight for search and rescue applications, to improve the speed, efficiency and safety of search and rescue to benefit both the victims and the rescuers alike. This system also alleviates the need for large teams of rescuers to divide up and search vast areas of land were a stranded victim could be. To accomplish this, the system was designed to incorporate light, long endurance UAVs, equipped with specialized search and rescue sensors to aid humans in the search for lost hikers in mountainous areas. The ability to search from the air, without putting additional humans at risk is invaluable for search and rescue. All UAVs in the system utilize the paparazzi autopilot system, which is an open source, Linux based autopilot package for flight stability and autonomous control. The system was engineered to follow a centralized command structure, revolving around a specially outfitted UAV, the mothership. The mothership communicates with the other UAVs, communicating supervisory tasks and coordinating the efforts of the UAVs to be as efficient as possible. The mothership also communicates with the ground station, where rescuers on the ground can relay commands to the network, and vital information can be passed down from the UAVs. To date, we have accomplished RC flight of the mothership, as well as an assisted autopilot flight of one of the drone models, where the autopilot was given control of the roll and the pitch while the pilot maintained control of the rudder and the throttle.

Executive Summary

The purpose of the executive summary is to allow it to serve as both an executive summary as well as an easy entry for future conference papers.

Design of an Autonomous Platform for Search and Rescue UAV Networks

Catherine Coleman, Joseph Funk, James Salvati, Chris Whipple
Taskin Padir, Alexander Wyglinski

Worcester Polytechnic Institute
100 Institute Rd Worcester, MA 01609
projectwind@wpi.edu

ABSTRACT

This project designed and implemented a platform for use in a system of unmanned aerial vehicles (UAVs) capable of human assisted-autonomous and fully autonomous flight for search and rescue applications to improve the speed, efficiency, and safety of search and rescue to benefit both the victims and the rescuers alike. To accomplish this, the platform was designed to be lightweight with long endurance, equipped with specialized search and rescue sensors, and utilizes the paparazzi autopilot system, which is an open source, Linux based autopilot package for flight stability and autonomous control. This project worked in conjunction with two other WPI MQPs, an AI/Image processing team and a communications team, as well as two teams from the University of New Hampshire, which built the communications hardware and UI, to realize the full system, including inter-UAV communications, high level search algorithms and ground control station with a user interface.

1. INTRODUCTION

Hiking is a very popular activity for people across the globe. Some hikers are leisure seekers, looking to get away from the sights and sounds of a city and escape to the great outdoors. Others are serious hikers, who are exhilarated by the danger and challenges presented by attempting to climb a 4000 foot tall mountain. Unfortunately, each year thousands of people in the US get lost or injured while hiking, and must be rescued by professional search and rescue teams. In fact, just in America's National parks, "from 1992 to 2007 there were 78,488 individuals involved in 65,439 [search and rescue] incidents" (Heggie, et al 2009). Of these, 3.39% ended up in fatalities and 30.94% in personal injuries to the victim. 3.39% is a small fraction, but this number would be closer to a 20% fatality rate without the presence and quick response of search and rescue teams (Heggie, et al 2009). This clearly shows, from the victim's perspective, the need for fast and accurate search and rescue efforts in our national parks and beyond. The challenges and risk to the rescuers are great as well. "Musculoskeletal injuries were the most common type of injury that SARA [Southern Arizona Rescue Association] members suffered over [30] years...These included two fractures, a disrupted knee, a shoulder dislocation, and sprains requiring emergency department evaluation. There were many more minor sprains and strains for which no assistance was needed" (Iserson, et al 1989).

Current search and rescue techniques include a lot of heavy machinery, including, but not limited to, cars, trucks, all terrain vehicles, helicopters and planes. The rough terrain compounds the need for skilled operators, pilots, and drivers. Pilots need to take planes and helicopters close to the treetops in order to be able to spot the survivor through the trees, combined that with the ever-changing slope of a mountain range, and it becomes a recipe for disaster. The task of searching is no easier for the ground crews, navigating cars and trucks on dirt paths and windy mountain roads, avoiding obstacles like canyons, fallen trees, rocks, rivers and cliffs. Eventually, over the course of a search, they have to leave the relative safety of their vehicles in order to venture further into the wilderness by foot, leaving them exposed to the elements anyways (Joint Chiefs of Staff, 1991). There is much room for improvement, which is where Project WiND can help.

Project WiND was created to develop a network of unmanned aerial vehicles (UAVs) for use in search and rescue applications. The idea is to send multiple, inexpensive, autonomous aerial vehicles out to fly low over the trees to aid rescuers find victims efficiently without risking the lives of rescuers. The goal being to cover vast amounts of area quickly, while streaming back data regarding potential victims back to a base station for verification by rescuers. Once a victim is identified, a search team can be dispatched to the exact location to extract the victim. This will limit the time the rescuers are actually out in the field and therefore help keep them safe. Also, the efficient, coordinated effort of multiple UAVs will greatly improve the amount of area which can be covered by a standard ground based search team.

Project WiND was divided into three separate projects: the platform team, the software team, and the communications team. The software team was responsible for the creation of the search algorithms to ensure that each UAV in the network was utilized effectively and efficiently. They also were responsible for creating the image processing and data processing necessary to limit the amount of unnecessary data being sent back to the rescuers on the ground. The communications team was

responsible for creating a resilient wireless link between all of the UAVs in the network as well as a link from the network to the ground station. This report represents the last team which is responsible for the design and manufacturing of the platform.

2. DESIGN SPECIFICATIONS

To accomplish these lofty goals, several design specifications were established based on existing research and the needs of the other teams working on this project. The list of design specifications included:

- Three fully autonomous fixed wing aircraft
- Interface with the UAV controlling search algorithm
- Autonomous navigation based on GPS waypoint navigation. The aircraft will need to be able to receive GPS coordinates from the search algorithm and then proceed to fly to the given coordinate then circle, awaiting for new GPS waypoints.
- Each UAV will need to carry a payload of 10 lbs., in order to carry all of the processors, cameras, flight sensors, and gimbal.
- Capable of flying at a cruising speed of 35 – 45 miles per hour under full load of search and rescue sensors
- Capable of flying between 120 ft. and 400 ft. of altitude as specified by the FAA for UAVs
- Capable of maintaining flight without refueling for a minimum of 1 hour
- Capable of supplying 102 watts of power to on board electronics, for a minimum 1 hour per charge
- Contains a pan tilts camera gimbal capable of holding up to two cameras, one color camera and one infrared camera.

These specifications are necessary metrics for the success of the project. For example, it is important for the flight time to be in excess of 1 hour. Otherwise, the UAVs will be unable to cover any significant area per fueling, which would make the system a hindrance as opposed to an asset to the search teams. Similarly, the power and maximum payload requirements are also important, and have been derived from the needs of the other teams of project WiND. With a system that meets these specifications, the network will be capable of flying up to 30 miles on a single flight. The efficiency of the search algorithm varies depending on the terrain being searched as well as other factors; however, this could translate to a round trip search of 38,250,000ft^2 at 400 feet in altitude per aircraft.

3. PLATFORM EXECUTION

After gaining an understanding of the problem we entered the execution phase of the project. During this phase we designed, built, and tested our UAVs.

3.3 PARTS SELECTION AND MODIFICATION

We choose to use the Senior Telemaster and two Skyline Champ airframes during this process. They were chosen for their stable flight characteristics, and large payload capacity. We choose to use a 20 cc gasoline engine for the Telemaster and a 30cc gasoline engine for the Skyline Champs. We choose to go with gas engines over nitro and electric because it was more cost effective for the flight times based on our specifications.

We added an aluminum frame and a plywood floor to the Senor Telemaster airframe. The aluminum was added to provide a hard mounting point and to protect the electronics, and to support our modified wings attachment points. The wings on the Telemaster were designed to be attached with large rubber bands, because of our large payload we added bolts and pegs. The plywood floor supports our camera dome, gimbal, and Xbee radio antenna.

Figure 1: Telemaster Airframe out at the Airport

Figure 2:Blue Skyline Champs Airframe out at the Airport

Our camera gimbal can support up to two cameras, a colored and infrared camera. The final implementation only included the color camera. This gimbal serves two main purposes, to increase the field of vision of the cameras and dampen the vibrations from the engine and propeller. Its main source of dampening is a layer of foam-based silicon isolating it from the plane, and a spring steel linkage system that absorbs more of the vibrations that can be amplified by the inertial properties of the camera.

4. PLATFORM TESTING

We traveled out to Tanner Hiller Airport seven times to test our planes, with a majority of the tests in February. We had multiple flights during each of these tests. We modified the airframe after each flight test to improve its flight characteristics. The Telemaster plane was damaged during the second, third, fifth, and seventh test, and needed to be repaired. On the final fight we lost control of its elevator and it flew into the trees, and has been damaged beyond repair. Due to our modifications we were able to salvage all of our electronics. However the Blue Skyline Champ successfully flew under RC control. The yellow Skyline Champ was never tested.

5. PLATFORM RESULTS

Based on information from the GPS we were able to confirm that we meet both our speed and altitude goals.

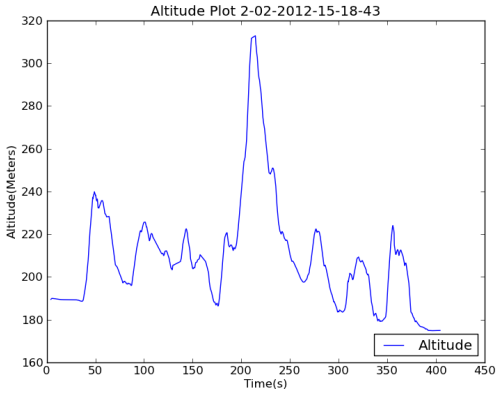


Figure 3: Altitude Plot from February 2nd Flight

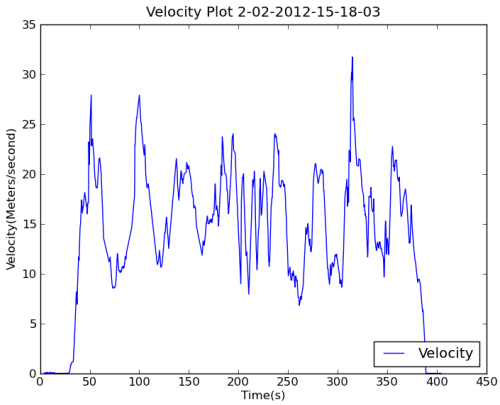


Figure 4: Velocity Plot from February 2nd Flight

By comparing footage from a camera hard mounted to the plane and the one mounted in gimbal shows that we were able to eliminate the sine waves in the video.

Figure 5: Video recorded with the Vibration Dampening

Figure 6: Video recorded without Vibration Dampening

6. CONTROL SYSYTEM

The control system used for the airframes we selected was called Paparazzi. Paparazzi is an open sourced, Linux based control system designed to autonomously fly fixed-winged airplanes and quad copters. For this project we will be implementing the autopilot system into our airframes with the goal of GPS waypoint navigation. This system is required to have the ability receive new waypoints from the software integration team, which is responsible for controlling the UAVs to follow a dynamic search and rescue navigation routine.

The autopilot system features three distinct modes, manual mode, Auto1 and Auto2. Manual mode is a direct control of the UAV via a radio controller. Auto 1 is a partial stabilization mode, which allows the user to configure which control surfaces the autopilot has control over to stabilize the plane to fly in a straight line. Auto 2 is a full autonomous mode which follows a flight plan that contains preconfigured flight blocks.

Paparazzi provides a ground control station which allows the user to view the UAV's approximate orientation, speed, altitude and pitch based on configured sensors. The team successfully configured a GPS module for position of the UAV and IR thermopiles for flight stabilization. The sensor data is transmitted wirelessly using Xbee radios from the UAV back to the ground control station for data recording and live feedback to the user.

7. CONTROLS SIMULATION

The ground control station provides the ability to run simulations based on predefined configurations for the autopilot control system. Running simulations allows for initial testing of the system to determine how it responds before traveling to the testing field. Two critical control loops of the control system is vertical and horizontal control. The vertical control loop was simulated by launching the plane and having it fly to a target altitude.

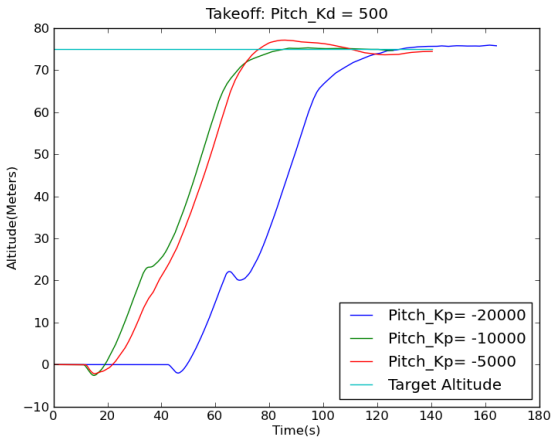


Figure 7: Simulation Results with different Kp values

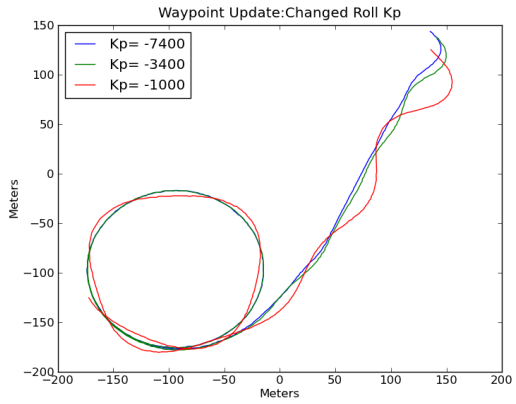


Figure 8: Simulation Results with different Kp values

By changing the proportional constant the control loop responsible for the position of the elevator, the change in pitch of the plane during ascent is affected. The horizontal control loop was simulated by setting a new target waypoint and examining the effect of the proportional constant on the roll set point of the plane. The roll set point controls the position of the ailerons, responsible for turning the plane.

8. CONTROLS RESULTS

On the senior Telemaster, the team successfully completed two complete manual mode flights. During one of the complete manual mode flights we successfully engaged Auto1 mode where the plane sustained a slight bank to the right. Future testing can straighten out the plane and then Auto 2 can be engaged to circle a target GPS position with a preconfigured radius.

Using the ground control station we can successfully connect two separate autopilot boards for data acquisition and autonomous control. This can be expanded to more autopilot systems to satisfy our multi-autonomous system design specification.

In order for our system to work with the software integration team, the autopilot control system must feature the ability to update waypoints from an external source. Using python, we successfully inserted new target waypoints into a paparazzi simulation. The waypoints inserted were generated from the path planning routine created by the software integration team.

Figure 9: Externally inserting waypoints

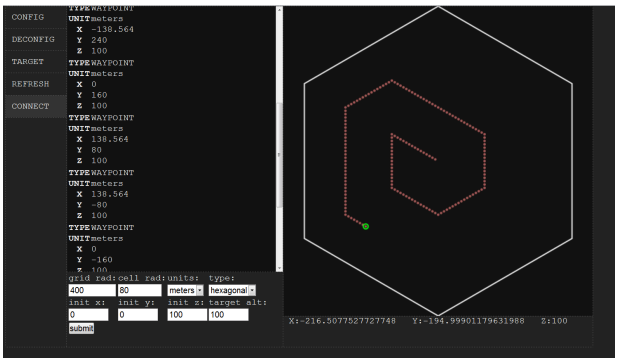


Figure 10: Software team's user control interface.

Figure 9 was taken during the paparazzi simulation where the plane is navigating in a spiral fashion from the center point. Figure 10 was taken from the software team's user interface with the outputted waypoints displayed on the left of the image.

9. CONCLUSION

Project WiND has made great strides towards the ultimate goal of creating a multi – UAV system for use in search and rescue. One of the drone models has successfully flown under assisted autopilot, the mothership has flown under radio control, and the second drone model is a few parts short of an RC flight. The autopilot boards are successfully connecting and communicating to a laptop over the Xbee radios, and multiple autopilot boards can connect to the same laptop simultaneously and stream data to the ground control software console on the laptop. Multi plane simulations are also capable in the ground station software.

9.1 RECOMMENDATIONS FOR FUTURE WORK

There is much room for the future expansion of this project. The first steps to future development need to start with finishing the assembly of the different airframes, as well as the installation and calibration of the autopilot systems in each. In the end, the system needs at least three UAVs capable of autonomous waypoint navigation. Once all of the UAVs are successfully flying under autopilot, the next step would be to integrate completely with the other teams.

Integration with the other projects would allow the users to fly the fleet of drones from the Microsoft Surface, update waypoints, and stream filtered images back from the different UAVs. To accomplish this, the software team needs to have an interface with the autopilot board in order to change waypoints as well as get current telemetry and location data. This process has been started, but an elegant solution has not been developed yet. Mechanically, the teams need to integrate all of the necessary hardware for each team into the UAVs, including the search and rescue cameras, FPGA board for image processing, the Wi-Fi modules, the power distribution boards, and the software team's logic board. All of these components need to be mounted in the UAVs with a vibration resistant mounting system and wired together. The layout inside the UAVs has been complete for all

of these components with the help of CAD models of the airframes to ensure that each airframe remains balanced with the addition of all the extra weight.

Once these have been complete, the system will be fully functional and ready to perform low altitude, efficient aerial search and rescue.

10. BIBLIOGRAPHY

National Search and Rescue Manual. Volume 1: National Search and Rescue System. [United States]: Joint Chiefs of Staff Washington Dc, 1991. http://www.towage-salvage.com/files/sar01/.

Heggie, Travis W, Amundson, Michael E. "Dead Men Walking: Search and Rescue in US National Parks". Wilderness & Environmental Medicine, vol. 20(3), pp. 244 – 249, 2009

Iserson, K V. "Injuries to Search and Rescue Volunteers. A 30-year Experience" The Western journal of medicine, vol. 151, Iss. 3, pp. 352 – 353 09/1989

Table of Contents

1. INTRODUCTION 1
2. PLATFORM DEVELOPMENT 8
2.1 Background 8
2.2 Platform Execution 15
Testing 33
2.4 Platform Results 39
3. UAV CONTROL DESIGN 43
3.1 Background 43
3.2 UAV Control System Integration 48
3.3 Autopilot Results 77
4. TIMELINE 83
4.1 Timeline with regards to the Hardware team 83
4.2 Timeline with regards to Integration 84
5. BUDGET 84
6. INTEGRATION WITH SOFTWARE AND COMMUNICATIONS 85
6.1 External Waypoint Insertion 85
6.2 Camera 88
7. RESULTS 89
8. CONCLUSION AND RECOMMENDATIONS 90
8.1 Recommendations for Future Development 90
9. ACKNOWLEDGMENTS 91
10. BIBLIOGRAPHY: 93

Table of Figures

Figure 1: This graph illustrates the breakdown of fatalities vs injuries vs unharmed survivors in all search and rescue operations in the United States from 1992 – 2007 (Heggie et al 2009). 1
Figure 2: This is an image of rescuers performing near vertical rock climbing to locate and attend to a victim, a very risky maneuver. (El Paso Country Search and Rescue) 2
Figure 3: The U.S. military's MQ-9 Reaper drone, used for counter terrorism armed attacks. (US Air Force, 2012) 4
Figure 4: The project organization flow chart. This illustrates how the different teams of Project WiND are organized and how they interact with each other. This paper represents the work done by the Hardware Platform team in the lower left corner. 5
Figure 5: Warbird, a type of scaled plane modeled after military aircrafts (http://www.raidentech.com/new20f4uco4e.html) 8
Figure 6: Sports Model, a type of scaled plane intended for aerobatic flight (http://product.madeinchina.com/RC-Airplane-Model-Sport-Fun-46P_13077777.shtml) 8
Figure 7: Sailplane, a type of scaled plane with long thin wings used for gliding 9
Figure 8: Airfoil Geometry, Flat Bottom 10
Figure 9: Airfoil Geometry, Symmetrical 10
Figure 10: Airfoil Geometry, Reflexed 10
Figure 11: Diagram of how to measure the wing dihedral 11
Figure 12: Lay out of a plane using the recommended ratios, based on a 10 unit wing span 12
Figure 13: CAD of Goose's design before we started construction. This CAD image doesn't include the MonoKote that will cover the wings and tail and fill in the gaps. 16
Figure 14: Laser Cutter template we used to cut out the airfoil in the wings 17
Figure 15: Fully assembled Goose, ready for flight 17
Figure 16: Unassembled Senior Telemaster Parts as they are shipped 18
Figure 17: Unassembled Skyline Champ Parts as they are shipped 19
Figure 18: Stock RCG 20cc Gas Engine out of the box 20
Figure 19 Internal layout with all of the components from all the teams 21
Figure 20 Internal layout with only the components necessary for flight 21
Figure 21: Robin with back hatches installed 22
Figure 22 Robin with the servos moved to the back of the tail 22
Figure 23: Gas tank in its original position all the way forward 23
Figure 24 Sony color block camera that we will put in the planes 23
Figure 25: Frequency responded for a silicon with a 6 on the Shore A hardness scale and optimal load of 0.2~0.75 (kg/ 4 legs) 24
Figure 26: Frequency responded for a silicon with a 10 on the Shore A hardness scale and optimal load of 0.5~2.5 (kg/ 4 legs) 24
Figure 27: Frequency responded for a silicon with a 14 on the Shore A hardness scale and optimal load of 4~15(kg/ 4 legs) 25
Figure 28 CAD of Camera Gimbal with Sony camera 25
Figure 29: Modifications to fit the gas tank chamber 26
Figure 30: The fuel tank chamber in place in the body 26
Figure 31: The quick connect values from inside the body of the plane, with reinforcing plate 27
Figure 32: The rubber grommets attaching the engine to the fuselage 27

Figure 33: Yellow Skyline Champ Duck, partially assembled 28
Figure 34: LM5118 efficiency curve 30
Figure 35: Standard configuration for buck-boost regulator 30
Figure 36: Configuration used in our schematic 31
Figure 37: Controller Circuit Diagram 31
Figure 38: Images from the video of Robin's Take off out at Tanner Hiller. 33
Figure 39: Images from the video of Jay's take off at Tanner Hiller 33
Figure 40 Gas tank in its final position moved 4 inches back 35
Figure 41 in side of Robin showing the different compartments 36
Figure 42 Aluminum support frame inside of Robin, with the wooden mounting plate for the wings 36
Figure 43: What remains of the bottom of Robin's fuselage after its February 18th crash. Notice how it is missing its landing gear and wing strut attachment points. The Aluminum frame prevented further damage. 37
Figure 44: Robin after its repairs from the February 18th crash. 37
Figure 45: Camera behind Robin's dome before its April 11th flight 38
Figure 46: What was left of Robin's body after the April 11th Flight. All of the electronics survived the crash. 39
Figure 47: Altitude Plot based on our February 2nd flight. 40
Figure 48: Velocity plot from our February 2nd flight 41
Figure 49: Camera image with the engine off, when the camera is mounted properly with vibration dampening 41
Figure 50: Camera images with the engine on, when the camera is simply taped to the bottom of the plane. 42
Figure 51: Autopilot Flow Chart. This chart represents the flow of how the autopilot system interfaces between the UAV in the air and the user controlled ground station. 43
Figure 52: Piccolo autopilot system. Piccolo is an all in one solution for autonomous flight provided by MIT. http://www.cloudcaptech.com/piccolo_II.shtm 44
Figure 53: Ardupilot is an open source autopilot solution featuring arduinos. 45
Figure 54: Paparazzi autopilot system. This particular version is named YAPA2. 46
Figure 55: IR Response Curve. This curve represents the response of the thermopiles as the angle of the plane is changed. http://paparazzi.enac.fr/wiki/File:Ir_response_curve.gif 47
Figure 56: IR Thermopiles. They are in charge of providing flight stabilization for autopilot mode. 47
Figure 57: The paparazzi configuration files allow for the complete configuration of the paparazzi autopilot system. This allows for many different types of airframe structures and sensor combinations to be implemented. 51
Figure 58 Paparazzi autopilot system showing the ground control station coordinator Xbee for connecting the ground station to the Xbee located on the Yapa2. 52
Figure 59: Adding in GPS subsystem. This line of code from within the airframe configuration file will add the mediatek_diy GPS module to the airframe. 52
Figure 60: The mediatek GPS located at the front of Robin. 53
Figure 61: Mediatek GPS located at the front of Jay. 53
Figure 62: The thermopile section within the airframe configuration file. This is where neutrals, signs and other parameters are configured. 54

Figure 63: Definition of the servos within the airframe configuration file. These definitions set the neutrals, minimum, and maximum positions for the control surfaces of the UAV. 55
Figure 64: Portion of the Telemetry configuration file. This shows some of the available types of messages that can be sent across the xbee radios along with the rate in which they are transmitted. 56
Figure 65: Above is an example of the messages window provided by the ground station to view incoming messages from the UAV. The “bat” message is currently selected and shows values such as voltage and current from the battery. 57
Figure 66: Above is a sample data file collected during tested. The messages above were selected by the telemetry file and then recorded when received from the UAV. 57
Figure 67: The top and the bottom of the MeekPe encoder board. This encoder board allows us to read the commands for the servo channels from the radio receiver without modifying it. 58
Figure 68: MeekPe PPM Encoder board block diagram. The encoder board takes in all of the servo commands and then sums them together into a PPM signal, which is easy to parse using the paparazzi radio configuration file. 59
Figure 69: Radio Control System for controlling the plane through the autopilot system. 60
Figure 70: Radio Configuration File. This file defines the structure of the PPM signal received from the MeekPe encoder board. 60
Figure 71: Waypoint Definitions within flight plan. These waypoints are example waypoints contained within the flight plan. The waypoints position is relative in meters from the home waypoint. 61
Figure 72: Visualization of the flight plan over Tanner Hiller, the local private airport the team tested the planes. 62
Figure 73: Wait GPS block within flight plan. This is an example of programmable blocks or steps within the flight plan. This example will run at the beginning of AUTO2 and until the GPS fix is valid. 62
Figure 74: Line navigation routine callback button. This block within the flight plan configuration file creates a function callback to the line navigation routine. 62
Figure 75: Settings Tab within Ground Station 63
Figure 76: GPS Message Structure. The GPS messages from simulations contain the position of the aircraft, which was used to examine the effect of changing the parameters of the control loops. 64
Figure 77: Paparazzi Autopilot Control Loop Structure. The overall structure features two main control loops, one for vertical, and one for horizontal, where the final output of each control loop define target positions for the control surfaces, driven by the servos. 65
Figure 78: Vertical Control Loop 66
Figure 79: Pitch Control Loop 66
Figure 80: Elevator Position Control Loop 67
Figure 81: Screen capture of the ground control station during a launch simulation 68
Figure 82: Pitch Launch Simulation where Kd = 0. This graph shows three different simulations where the kd constant was kept at 0 and then the kp constant was varied as represented in the graph key. 69
Figure 83: Pitch Launch Simulation Kd = -500. The graph above shows three simulations where the kd constant was kept at -500 and then the kp constant was varied. 69

Figure 84: Pitch Launch Simulation Kd = 500. The graph above shows three simulations where the kd constant was kept at 500 and then the kp constant was varied. 70
Figure 85: Horizontal Control Loop 71
Figure 86: Course Control Loop 71
Figure 87: Roll Control Loop 72
Figure 88: Waypoint Navigation Simulation 72
Figure 89: Waypoint Navigation Simulation. The goal of this simulation was to examine the effect of changing the roll_attitude_gain on waypoint navigation. When the plane was stable circling the standby waypoint the target waypoint was moved down the runway. 73
Figure 90: Flight Gear Simulation. This multi-image figure depicts the takeoff simulation of paparazzi alongside flight gear. Flight gear is an open source radio control flight simulator. The time of each picture was taken from the video is labeled. 74
Figure 91: Multi-plane Simulation. This multi-image figure demonstrates a simulation in which two separate planes are flown. One plane is in blue and the other is in red. The time in which the snapshot was taken is located on the bottom right of each picture. 75
Figure 92: Data Acquisition during Manual Autopilot control. This portion of data shows RC messages with complete RC commands from the remote controller and updated GPS positions. . 78
Figure 93: Autopilot data acquisition during RC flight. This figure alternates between the ground control station displaying the UAV's position and the UAV flying. 79
Figure 94: Successful takeoff using the manual mode of the autopilot. 80
Figure 95: Multi-plane connection. This image demonstrates connectivity between two separate autopilot boards. 81
Figure 96: Data Acquisition during Manual Autopilot control. This portion of data shows RC messages with complete RC commands from the remote controller and updated GPS positions. . 82
Figure 97: The General Gantt chart as predicted by the team in the beginning of October 2011 ... 83
Figure 98: Portion of the waypoint insertion script. This portion of code written in python inserted a target waypoint in x,y meters relative to home into a paparazzi instance. 86
Figure 99: Waypoint Insertion Simulation. This sequence of images demonstrates the insertion of the software team’s navigation routine into a paparazzi simulation. 87
Figure 100: Software integration team's GUI used to view the path of the UAV during a search and rescue mission. 88

List of Tables

Table 1: this table Identifies which flight characteristics are effected by different Design Parameters 10
Table 2: Power board specifications 29
Table 3: Test day summary results based on outcomes 34
Table 4: Shows the different types of support that we completed preparations for on the different airplanes 40
Table 5: Shows the different types of support that we Implemented on the different airplanes 40
Table 6: Decision matrix for determining which autopilot system the team used. 49
Table 7: Decision between using a paparazzi autopilot board verses using a development board. 49
Table 8: Decision between which paparazzi autopilot board to use. 50
Table 9 : Autopilot goals. This table lays out the autopilot goals and whether they were achieved, partially achieved, or not met. 77
Table 10: Summarizes the how successfully we accomplished all of our Goals during this project. 89

1. Introduction

Hiking is a very popular activity for people across the globe. Some hikers are leisure seekers, looking to get away from the sights and sounds of a big city and escape to the great outdoors. Others are serious hikers, who are exhilarated by the danger and challenges presented by attempting to climb a 4000 foot tall mountain. Unfortunately, each year thousands of people in the US get lost or injured while hiking, and must be rescued by professional search and rescue teams. In fact, just in America's National parks, "from 1992 to 2007 there were 78,488 individuals involved in 65,439 [search and rescue] incidents" (Heggie et al 2009). Of these, 3.39% ended up in fatalities and 30.94% in personal injuries to the victim, as visualized in Figure 1.

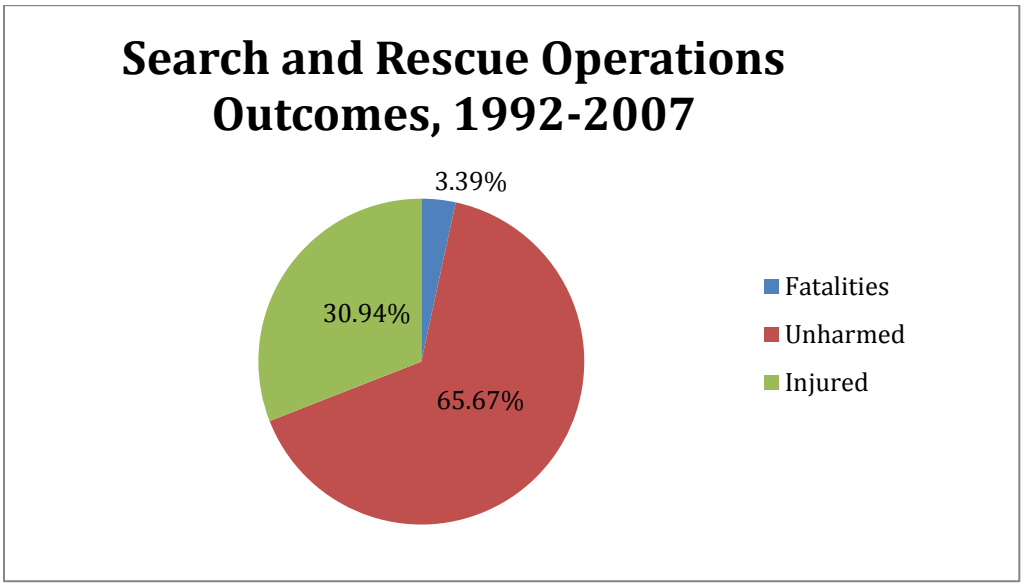


Figure 1: This graph illustrates the breakdown of fatalities vs injuries vs unharmed survivors in all search and rescue operations in the United States from 1992 – 2007 (Heggie et al 2009).

Figure 1 illustrates that only a small fraction of the victims in need of search and rescue perish as a result of being lost, but experts say that this number would be closer to a 20% fatality rate without the presence and quick response of search and rescue teams (Heggie et al 2009). This clearly shows, from the victim's perspective, the need for fast and accurate search and rescue efforts in our national parks and beyond. From the perspective of the rescue teams, the challenges and risk to the rescuers are great as well. "Musculoskeletal injuries were the most common type of injury that SARA [Southern Arizona Rescue Association] members suffered over [30] years...These included two fractures, a disrupted knee, a shoulder dislocation, and sprains requiring emergency department evaluation. There were many more minor sprains and strains for which no assistance was needed" (Iserson et al 1989). The Southern Arizona Rescue Association is a good example of the risks rescuers face since they are exposed to a wide range of search and rescue scenarios, including terrains like mountains, bodies of water and dessert rescue. Other injuries to rescuers included hypothermia and near drowning. (Iserson et al1989)

Calling in a search and rescue team is also quite costly; in the United States, almost $60,000,000 was spent between 1992 and 2007 on search and rescue missions in America's national parks (Heggie et al 2009). The high cost associated with search and rescue is due to the need for such large numbers of skilled people, specialized equipment, and the most important of resource of all, time. Search teams often have a wide search area to comb through looking for the lost or injured hikers. Current search and rescue techniques include a lot of heavy machinery, including, but not limited to, cars, trucks, all terrain vehicles, helicopters and planes. Even under the best conditions, these pieces of equipment require some skilled humans to pilot and/or drive; however, the rough terrain compounds this need for skilled operators, pilots, and drivers. Pilots need to take planes and helicopters close to the treetops in order to be able to spot the survivor through the trees, combined that with the ever-changing slope of a mountain range, and it becomes a recipe for disaster. The task of searching is no easier for the ground crews, navigating cars and trucks on dirt paths and windy mountain roads, avoiding obstacles like canyons, fallen trees, rocks, rivers and cliffs, as well as performing high risk maneuvers such as rock climbing, as depicted in Figure 2. This not only puts the rescuers at risk, it also takes a lot of time and requires a lot of training.

Figure 2: This is an image of rescuers performing near vertical rock climbing to locate and attend to a victim, a very risky maneuver. (El Paso Country Search and Rescue)

1.1 How is Search and Rescue done now?

Overall, whether on the ground or in the air, search and rescue is a dangerous occupation. It is also highly inefficient, because it relies on the trained eye of a human, who may or may not have a clear view of the area around the victim. Speed, efficiency, safety, and even cost, are all areas to be addressed in search and rescue operations.

According to the International Aeronautical and Maritime Search and Rescue Manual (IAMSAR) (Joint Chiefs of Staff, *1991*), all search and rescue missions are executed in five precise stages, each of which highly depend on the people in distress, their condition and the environment in which they are located. The first stage of any search and rescue mission is the awareness stage. Before any rescue attempts are made, there must be notification that there is a rescue to be made. Once a distress call is received by a local search and rescue (SAR) organization, the team next moves in to the initial assessment stage. The initial assessment consists of determining what type of incident has occurred. There are three different types of incidents which are uncertainty, alert, and distress, each which require different tactics. Following the initial assessment, there is a comprehensive planning stage. The goal of the planning stage is to derive the fastest and safest way to find the lost hiker and to coordinate the many different teams of rescuers once the plan is being executed. The planning stage requires skilled rescuers, typically with intimate knowledge of the area to be searched. After the planning stage, the operation begins. This is when the rescuers risk their lives to search the area for

the missing hiker. Once the operation is completed, the mission comes to the conclusion phase when the mission is considered successful or unsuccessful and all personnel return from the mission. (Joint Chiefs of Staff, 1991)

Speed is crucial with regards to the awareness and initial assessment stages, as the chances for survival of an injured person decreases by as much as 80% in the first 24 hours. The urgency of the response is dependent on the terrain, climate, ability, and endurance of the survivors. Regardless of the situation, it is to be assumed that all of the survivors are incapacitated and require assistance as soon as possible.

When the location of the survivors is unknown, a well-developed search plan is pivotal to the success of the mission. The first step to making a search plan is to first evaluate the situation, taking into account elements such as the intended route of the victim, their last known position, any hazards along the path of travel, the experience level of the victim, the current and near future conditions of the search environment and the results of any previous searching. The second step is to estimate the distress incident location. The distress incident location is estimated by taking the information gathered about the victim from the first step and using it to determine the greatest distance the victim could have traveled from their last known position. Once they have an estimated location, the next step is to determine the best way to use the available search personnel to obtain success. This is a critical step as this will determine the rescue teams starting location and where they will concentrate their search path. Once the plan has been determined it is administered to the rescue team and executed. (Joint Chiefs of Staff, *1991*)

For initial searches, the higher amount of resources available, the greater the success of the mission and the sooner the distressed victims are found. Directly affecting the success of the mission is the sweep width of the SAR resources used. The sweep width is how much area a search team or equipment can search per pass. Sweep width is affected by the terrain, the object being searched for, and the type of resource. For helicopters and slow fixed-winged crafts, the ideal search height is 500 ft, allowing for a wide sweep path while not losing resolution of the search area. For maritime search and rescue, it is considered impractical to even attempt to spot a person in open water from greater than 500 ft. For teams on foot, sweep width is determined by the visibility of others on ground. If a rescuer cannot see the rescuer to their right, they are too far away. This is heavily impacted by the environment in which the rescue is performed – a dense forest will require smaller sweep widths than an open field.

1.2 How can this be improved?

Unmanned Aerial Vehicles (UAVs) can be used to reduce costs while increasing the amount of area searched. A UAV is an aircraft that is controlled autonomously or remotely from a ground station, and therefore does not require a human to physically be on board to pilot the aircraft or work the various sensors and tools onboard. This allows the aircraft, which can be a fixed wing airplane, helicopter, quad-rotor helicopter, flying wing, or any other airborne platform, to be smaller and therefore more fuel efficient and cheaper to build, buy, transport, and replace. Through the use of UAVs in search and rescue, it is possible to decrease the amount of resources needed in the initial

search and still increase the chance for success. This is where our project plans to improve search and rescue, using an autonomous UAV network to create and execute a methodical search pattern from the air, thereby limiting the number of rescuers put in harm's way.

1.3 How UAVs are being used

UAVs are multi-purpose vehicles that have been constantly evolving since their creation. They have a wide variety of roles from scientific exploration to armed military attacks. There is only one significant difference between a UAV and traditional aerial vehicles, which is the human interface. Traditional aerial vehicles need to carry a human on board to function as a pilot. This then requires a great deal of support for the pilot and crew aboard a vehicle, including space for the people to sit as well as life support systems. All of this adds weight to the aircraft, requiring a large aircraft to lift the extra weight. If you remove the human from the aircraft, there is room for more capabilities. Without the human on board, flight times can be unlimited, the airframe can be smaller, and more sensors, computers, weapons, or whatever the role of the aircraft requires can be added.

One of the most critical roles for UAVs is remote sensing and surveillance. UAVs were originally used as reconnaissance drones in high risk areas as early as World War I (Naughton 2003). Since then, technology has been advancing. New sensors have been developed and implemented, both for flying the UAV as well as for reconnaissance. The most common sensor on a UAV is some variant of a camera, a rather simple but very valuable sensor. There is currently a wide verity of sensors that are being used on UAVs such as atmospheric sensors, geomagnetic sensors, internal measurement units (IMUs) and synthetic aperture radar, just to name a few. With this great variety of sensors, the exact use of UAVs is almost unlimited. Some common uses of UAVs include:

- Remote Sensing (Reuder, Joachim et al, 2009)
- Commercial Aerial Surveillance (Jensen, A.M et al, 2008)
- Scientific Research (Reuder, Joachim et al, 2009)
- Armed Attacks (Bolkcom, 2004) (Schmitt, 2012)
- Civil Defence / Border Patrol (Bolkcom, 2004)
- Search and Rescue (Michael A. Goodrich et al 2008)

Figure 3: The U.S. military's MQ-9 Reaper drone, used for counter terrorism armed attacks. (US Air Force, 2012)

Figure 3 is an image of an MQ-9 Reaper UAV currently deployed by the United States Air Force and is used as a hunter/killer drone armed with AGM-114 hellfire missiles. Its current primary mission is for use in counterterrorism missions (U.S Air Force, 2012). The use of UAVs in search and rescue is not common yet, but with the increasing advances in technology, and decreasing costs, it may not be too far off in the future that systems like the one developed by this project will be commonly employed in search and

rescue missions. The great benefit of these drones is that they will allow a great amount of area to be searched in little time with a small amount of man power, all while keeping rescuers safe.

1.4 Project Division

This project was committed to filling the need of search and rescue operations by providing a system that will successfully search large areas, on the order of tens of square miles, in a small amount of time, on the order of hours, without putting human rescuers in harm's way. Designated Project WiND, the main project has these lofty goals. To make this more manageable, the project was divided into three sub-projects at Worcester Polytechnic Institute (WPI) and two additional projects at the University of New Hampshire (UNH). The three WPI projects included the software development team, the communications team and the platform design team. The UNH teams were a human interface design team and a second communications team with a focus on the radio hardware. Figure 4 is the project organization flow chart which visualizes the team organization and how the five teams will interact.

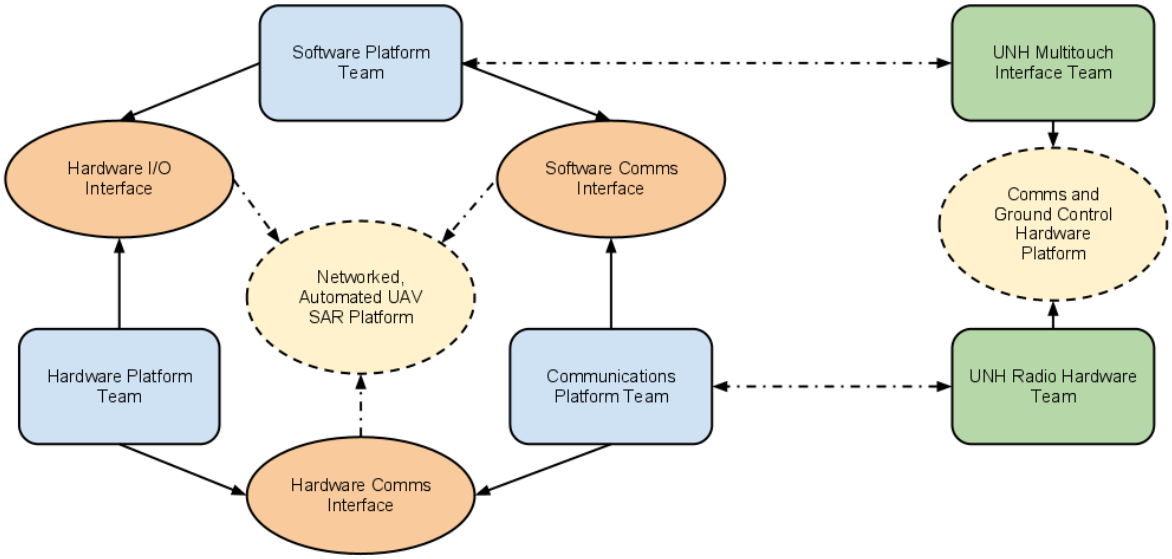


Figure 4: The project organization flow chart. This illustrates how the different teams of Project WiND are organized and how they interact with each other. This paper represents the work done by the Hardware Platform team in the lower left corner.

The software team at WPI was responsible for designing an algorithm to accurately and efficiently control the real world locations of the UAVs with respect to each other and the terrain being searched. The algorithm uses a Google maps image of the area being searched, and applies a matrix of hexagons and overlays the image with the hexagons. These hexagon cells are then given values using a probabilistic map, which is based off terrain data as well as information about the victim, including their last known location. Each cell is determined by basic assumptions of what the victim would statistically do. For example, it is statistically unlikely that the person would climb to higher altitudes once they become lost. The algorithm takes this statistic into account, and gives a higher priority to the terrain at or below their last known position. The software team was also actively involved with

writing algorithms to perform image processing for the on board cameras. A field programmable gate array (FPGA) was used to run the image processing algorithm, which primarily preformed color filtering and a blob detection algorithm to filter out the unnecessary frames from the camera. The algorithms detected anomalies in the image which could potentially be humans, and streams the possible hits to the base station for verification by a rescuer on the ground. The main goal was to filter out all of the images that do not depict humans to reduce what would otherwise be an enormous amount of data from the cameras to a much more reasonable amount that a rescuer could easily observe and verify the UAVs findings.

The WPI communications team was responsible for creating a link between the different UAVs and the ground station. They ultimately used a USRP2 software defined radio as the primary link between the UAV network and the base station. To use the USRP2 effectively, a computer capable of running Simulink was required to fly with the radio, since Simulink is used to control features such as the frequency, baud rate and syncing the two radios. For communication between the UAVs, a simple ad hoc wifi network, IEEE 802.11 standard, would connect the UAVs within the system. The UNH communications team designed and built the actual radio to be used between the UAV network and the ground station.

The second team from UNH was responsible for creating a human interface utilizing a Microsoft Surface to create a large touch screen based graphical user interface (GUI) to control the UAVs. This is the actual interface which works as a mask to the underlying search algorithm defined by the software team. The concept is to allow the rescuers on the ground to alter the flight plans of the UAVs due to new information about potential locations of the victim. This is also where the filtered images from the on board search and rescue cameras are displayed and analyzed by the rescuers.

Bringing all of these efforts together is the final portion of the project, the platform development team, whose work is represented by this paper. The platform team was responsible for designing and building a fleet of UAVs capable of autonomous flight as well as carrying the large assortment of radios, sensors, and onboard computer processors.

The layout of the network is such that there are two different types of UAVs in the network: the UAV designated the mothership, and the rest are considered drones. The mothership is the central communications hub, both between all of the UAVs and the ground station. The mothership carries the central path-planning computer which runs the path-planning algorithm developed by the software team. The mothership uses the software defined radios developed by the communications teams at UNH and WPI to talk to the ground station, and as such requires radios and all their supporting hardware, including a small Micro ATX computer and appropriate amplifiers and antennae. This allows the people on the ground to alter the search patterns of the UAVs while they are in flight. Data is streamed to the mothership from the ground station, the changes to the path are calculated, the most efficient path is selected, and then the path is dictated to the other drones in the network. All UAVs carry search and rescue sensors, and are autonomously controlled by the same autopilot system, with the same sensors.

1.5 Design Specifications

To accomplish this mission several design specifications were established based on existing research and the needs of the other teams working on this project. The list of design specifications included:

- Three fully autonomous aircraft
- Interface with the UAV controlling search algorithm
- Autonomous navigation based on GPS waypoint navigation. The aircraft will need to be able to receive GPS coordinates from the search algorithm and then proceed to fly to the given coordinate then circle, waiting for new GPS waypoints.
- Each UAV will need to carry a payload of 10 lbs, in order to carry all of the processors, cameras, flight sensors, and gimbal.
- Capable of flying at a cruising speed of 35 – 45 miles per hour under full load of search and rescue sensors
- Capable of flying between 120 ft. and 400 ft. of altitude as specified by the FAA for UAVs
- Capable of maintaining flight without refueling for a minimum of an hour.
- Capable of supplying 100 watts of power to on board electronics, for a minimum 1 hour per charge
- Contains a pan tilts camera gimbal capable of holding up to two cameras, one color camera and one infrared camera.

These specifications are necessary metrics for the success of the project. For example, it is important for the flight time to be in excess of 1 hour. Otherwise, the UAVs will be unable to cover any significant area per fueling, which would make the system a hindrance as opposed to an asset to the search teams. Similarly, the power and maximum payload requirements are also important, and have been derived from the needs of the other teams of project WiND. With a system that meets these specifications, the network will be capable of flying up to 30 miles on a single flight. The efficiency of the search algorithm varies depending on the terrain being searched as well as other factor; however, this would translate to a round trip search of 38,250,000ft^2 at 400 feet in altitude per aircraft. The success of the project was determined based on the completeness of these specifications at the end of the project.

There is a need for a more efficient, easier, and safer way to search for lost and injured hikers in mountainous regions. Our proposed network of UAVs will ultimately create a safe and efficient way to search vast amounts of wilderness without putting rescuers at risk, while still giving the victims the assistance they require.

2. Platform Development

The first portion of this project was to develop a platform capable of both autonomous flight and supporting higher level functions for the network. This section is dedicated to covering the background, the approached used, testing, and results for the platform development.

2.1 Background

Before attempting this project the team needed a though understanding of the current state of the art for designing, building, powering, and flying scaled planes. Additional solutions were researched including buying kit RC planes to manufacturing a custom solutions. This background information heavily influenced our final decisions.

2.1.1 Airplanes

An airplane is only as good as its airframe. The airframe determines how well it will fly and restricts its speed and flight time. Having a good airframe with an appropriate shape for our project is an important first step. In the following section the different types of airframes are discussed in addition to the basics of aircraft design and construction.

Types

There are many styles of airplanes each one has its own advantages and disadvantages. Using the correct style planes in this project can make both performing search and rescue patterns easier, and collecting visual information simpler.

Trainers are specifically designed for beginners. They are usually larger models made from balsa wood, with the wings mounted high on the body (Carperner 2002). They usually fly slowly and will return to straight flight, level flight, when not being directed (Tower 1994).

Scale models recreate full-size models. These are for advanced RC builders and flies, and are difficult to assemble and fly.

Figure 5: Warbird, a type of scaled plane modeled after military aircrafts (http://www.raidentech.com/new20f4uco4e.html)

Warbirds are planes that are modeled after military aircrafts. These planes are smooth flying, and aesthetically pleasing (Weis 2007). Some of them are also modified to improve their flight characteristics.

Sports Models are typically narrow wing airplanes intended for aerobatic, and for advanced fliers. Most of these planes have the wings mounted near the bottom of the body. They exchange stability for maneuverability.

Figure 6: Sports Model, a type of scaled plane intended for aerobatic flight (http://product.madeinchina.com/RC-Airplane-Model-Sport-Fun-46P_13077777.shtml)

F**latOuts** planes are typically made from re-enforced foam. They are quick and easy to assemble and easy to fly. However they have short flight times because they are unable to

support large power plants. Usually used to practice aerobatic maneuvers or for spontaneous flights.

"**Park fliers** are small electric powered airplanes that can be flown just about everywhere"(Towers 1996). They are cheap and durable, since they are made from plastic, instead of balsa wood. They also can be flown safely anywhere. They are also exclusively for beginners.

Sailplanes have long thin wings and are used to glide, and rise using the atmospheric thermal streams. Because of how easier they are to glide, they are used to travel long distances slowly. They can also support quite a bit of weight. While a very stable platform, this type of plane is difficult to maneuver.

Figure 7: Sailplane, a type of scaled plane with long thin wings used for gliding

Designing an Airplane

A well designed airplane requires a lot of time and thought. According to Paul Johnson, a model aircraft designer, there are several clear steps to designing a small aircraft (Johnson 2007).

Specifications

Johnson first recommends determining the specifications for the vehicle. These specifications may include any of the following and are discussed in detail in the following sections:

- Purpose of the model
- Type
    - Trainer, Scale, War bird, park Fliers, etc.
- Flight time – Based Size and Propulsion choice
- Stability
    - Should the model be self-stabilizing, neutrally stable or somewhere in between?
- Airspeed envelope
- Vertical performance
- Control response
- Stall characteristics
- Construction methods
    - Traditional wood, composite, etc.
- Control system
- Landing gear system
- Break-down for transportation

Propulsion

The next step is to choose a propulsion source. Selection of a propulsion source is important because if an airplane is designed to support a range of different power plants it must be structurally designed to support the largest one. This might not be the best aerodynamically or end up too heavy for the smaller power plants to lift. The speed of the airplane is equal to the product of the propellers rotational speed and the pitch of the propeller.

Vertical Performance and Airspeed

Step three is to determine the Vertical Performance and Airspeed. The rate of climb is a factor of the power plant, propeller being used, and the ready to fly weight. The propulsion source, propeller and wing design determine the airspeed. Both of these will affect the size and payload of the final airplane design. The rate of climb may be compromised in the final design to achieve the desired speed.

Wings

The next step is to design the wings. There are several different components to designing the wings. The chart indicates which aspects of flight each wing factor effects.

Table 1: this table Identifies which flight characteristics are effected by different Design Parameters

Flight Characteristic	Design Parameters					
	Airfoil	Wing Loading	Aspect Ratio	Dihedral	Washout	Aileron (Area/Style)
Airspeed	✓	✓	✓			
Roll Rate			✓	✓		✓
Stall	✓	✓	✓		✓	
Stability				✓	✓	
Lift Capability	✓	✓	✓			
Lift/Drag Ratio	✓	✓	✓	✓		
Aerobatics	✓	✓	✓	✓	✓	✓

The Airfoil

"The wing geometry affects the wing lift and center of inertia distributions in a mostly intuitive way" (Stanford University 2001). They come in several different styles (Johnson 2007): symmetrical, semi-symmetrical, flat bottom, modified flat bottom, under cambered, and reflexed. Each one of these styles contains a large number of airfoils that have similar shapes. Also data on real airfoils don't directly apply to their scaled down counter parts. However they do scale down comparatively. Symmetrical airfoils are for aerobatic airplanes. Semi-symmetrical airfoils are for secondary trainers, sailplanes and sport aerobatic biplanes. Flat bottom and modified flat bottom airfoils are used for slow gentle flight. Under cambered airfoils are used for scale models and sailplanes. Reflexed airfoils are used for fling wings. When picking an airfoil it is important to know the amount of lift, stall characteristics, leading edge radius, and intended purpose that each airfoil/style has.

Figure 9: Airfoil Geometry, Symmetrical

Figure 8: Airfoil Geometry, Flat Bottom

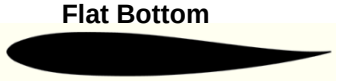


Figure 10: Airfoil Geometry, Reflexed

Wing Loading

Wing loading is the measurement of weight carried by each unit of area. The lower the wing loading the slower the plane will lift-off, fly and land. The higher wing loading the more predictable the plane is when landing, less responsive to controls, and must fly faster to stay in the air. Wing loading is typically measured in oz/ft (Johnson 2007), and includes the surface area of both wings. This number

in combination with the expected ready to fly weight can be used to calculate the appropriate wing area.

Aspect Ratio

The aspect ratio is the ratio of the wingspan to its chord (or width). This ratio affects the roll rate, lift-to-drag ratio and pitch sensitivity. This means that it also affects fuel efficiency and the planes tendency to tip stall. However if the aspect ratio is too high it will have sluggish response to rolling and is easier to break. A well-designed airplane compromised between having a large aspect ratio and being less maneuverable, and having a low ratio that is twitchy and inefficient. A good aspect ratio to aim for is 5:1(Tim 2001).

Dihedral

Dihedral is an angle that raises the wing tip above wing root. Giving a plane a dihedral increases its stability and its ability to prevent rolling. However it can also add an undesired control coupling. Control coupling is when the airplanes tilt along an unwanted axis, when using the rudder. "Too little dihedral will make turns sluggish. Too much dihedral will make the wing inefficient" (Johnson 2007). A dihedral can be used to turn a plane without ailerons and only a rudder.

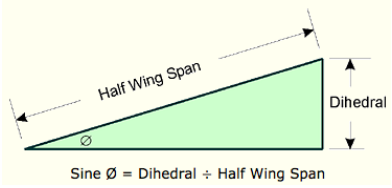


Figure 11: Diagram of how to measure the wing dihedral

Washout

Wings are built with a twist, so that the tips are at a lower angle incident than the wings root. This avoids tip stalls but also limits the planes aerobatic capabilities. This also reduces structural weight. These are useful in combination with high aspect ratios, heavier planes, and other planes not intended to perform aerobatics.

Aileron/ Flaps

Ailerons are the control surfaces on the wings. Ailerons are usually 10%-20% of the total wing area. These control surfaces are used in turning and rolls. Flaps are used to change the geometry of the wing in order to increase lift. This is used to allow the plane to make a slower approach during landings. For some RC aircraft, the ailerons can be used as flaps when properly configured in the transmitter or on board autopilot.

Lay it Out

The next step is to start drawing reference lines on paper, and lay out the general location of things to achieve the correct proportions and moments. The fuselage length should be around 75% of the wingspans. The nose length should be around 20% of the fuselage length. The horizontal tail surface area should be around 25-30% of the wing area. The vertical tail should be around 35% of the stabilizer area (Johnson 2007). Figure 12 illustrates a plane with the ideal ratios. Next it is recommended to use a computer aided design (CAD) program to model all of the components and balance the airplane. This CAD model should be an accurate representation of the final product.

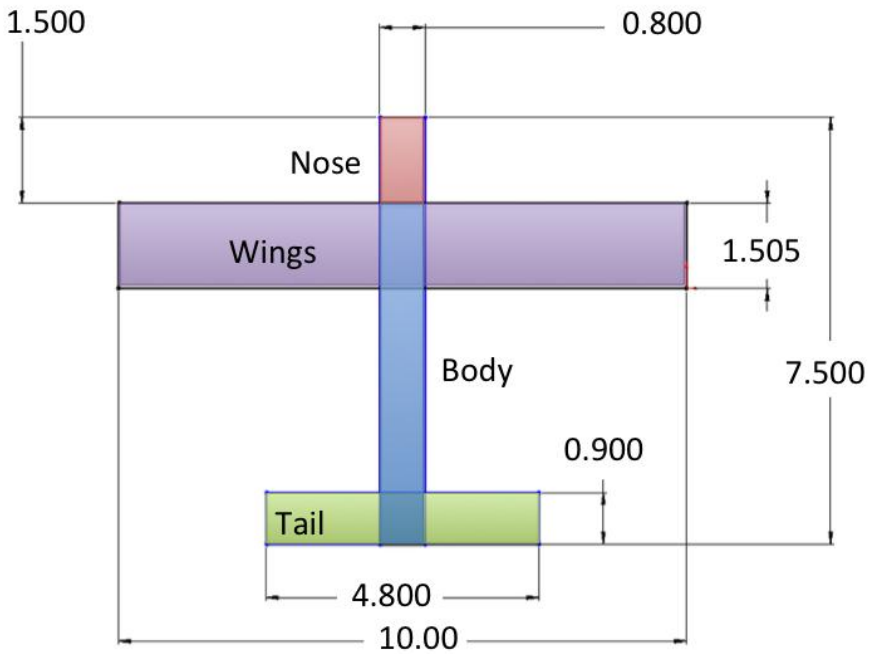


Figure 12: Lay out of a plane using the recommended ratios, based on a 10 unit wing span

Construction Techniques

There are several different materials and techniques commonly used for building homemade aircrafts. Each of these methods have their own unique advantages and disadvantages, and targeted application. The most common building materials are wood, fabric, plastic, metal, foam, and composites.

Balsa Wood

The advantages of working with a wood frame are that it requires only simple tools for construction and are fairly inexpensive to build. They also don't demand the close tolerances that other materials do. Considering the size and weight of the aircraft the project calls for, balsa wood is a good choice. Balsa wood is light, readily available, and cost efficient. Usually when constructing a wooden frame, the basic shape and airfoils are designed then modeled by a ribbed structure that will support the covering in the desired shape. These rips can be cut out by hand or using a laser cutter then attached together using some type of epoxy or wood glue. Once the frame is assembled it can then be covered in cloth, plastic or more wood. For full scale aircrafts, like personal airplanes, fabric made from high-grade polyester is usually used to cover the frame. However for small crafts like UAVs and RC airplanes, fabric is simply too heavy for the amount of durability it provides. For lighter planes, a plastic covering called MonoKote is often used to cover the frame. Heat is applied to shrink it and seal it around the structure. For aircrafts that need a more durable covering, lightwood is sometimes used to cover the structure. Wet wood is malleable enough to be warped around the frame while keeping its hardness when re-dried. While wooden frames are easier to build and many times more cost effective, they are not nearly as durable or reliable as other materials.

Foam

Foam is commonly used in small aircraft construction (Smith 2006). Foam is a very ridged and compression resistant material for its weight. It is also a very easy material to work with since it can be

easily shaped by either a hot knife or hot wire. These planes are often creating using sheets of Polystyrene (Blue), Polyurethane, PVC Foam, or Honeycomb foam. Polystyrene is most people's first choice and is commonly used in the wings. However, gas and other solvents dissolve it. Polyurethane is commonly used in fuselage construction because it is resistant to solvents. PVC foam is also resistant to most solvents, in addition to being able to withstand high temperatures. The advantage of this building technique is that it ends up with very smooth surfaces at a low cost.

Metal

Another material used when building homemade aircrafts is metal. These planes are more challenging to build, and require metal cutting and shaping tools. There are three main types of metal construction: sheet aluminum, tube aluminum or welded steel tubes. Planes created out of tubes are similar to wooden aircrafts and are usually covered in aircraft fabric. However since they are made from metal they don't require as complicated a support structure, in addition to being harder to damage. Sheet metal aircrafts are popular because they are easy to construct and fairly light for their durability.

Composite Materials

Another common practice is to make a composite material frame from a mold or by rapping a foam part (Alexander 1997). Composite is usually only used on larger airplanes because of how heavy the final product is. Composite airplanes consist of a core material, a reinforcement material, and a resin.

Core materials are most commonly made from foam however other materials like wood and honeycomb are also used. The most common foam used in airplane construction (Polystyrene) can't be used with polyester/vinyl resins. This material provides the shape of the part and supports it during the curing process.

There are three main types of reinforcement materials; fiberglass, carbon fiber, and Kevlar (Harris, 2005). Fiberglass is the most widely for reinforcing the size planes that will be used in this project. It also comes in various weights. While e-glass fiberglass is the most common, s-glass fiberglass, which is 30% stronger and 2-3x the cost, is also used. Carbon fiber is the strongest. Some negative qualities of these materials include how difficult it is to use, and how useless it is after breaking.

The last material used is Kevlar. This material is the least popular since it is difficult to work with and has a low compression strength. The advantage of using this material is that it is tensile strength. All of these different reinforcement materials come in either unidirectional or bi-direction weaves (Alexander 1999). Bi-directional weaves are usually more expensive because they can handle forces coming from more directions so they are less likely to break, but not necessary in every application.

The last component to a composite airframe is the resin, including polyester, epoxy, and vinyl. Polyester is the cheapest type and hardens the fastest. Unfortunately it is not appropriate for high

strength applications and shrinks over time, which makes it the least suitable for aircraft construction. Epoxy is the strongest and lightest resin of the three, however it needs a skin protection because it is heat sensitive. Its varying drying times can also make it difficult to work with. Vinyl resin, which is a cross between polyester and epoxy, has the best properties of both materials. It is both strong and flexible. It also doesn't have any moisture issues like the epoxies. Like the Polyester material it is easy to use, inexpensive, and fast drying. The only negative to this option is that it requires precise mixing, and can result in an explosive solution.

When creating a composite aircraft the first step is to mold the core material into the desired shape. Next wrap it in the reinforcement material, then cover the reinforcement material with the resin until it is thoroughly saturated (Honus 2009).

2.1.2 Propulsion

The three primary power sources for propulsion used in modern aircrafts of this size is electric motors, gas engines, and glow engines. All of these systems have their own advantages and disadvantages, and would work for our situation.

Electric

Using an electric motor to power an airframe allows for an easy simple solution that requires minimal maintenance. The biggest downside to this is the source of power. Electric motors rely on batteries that are heavy. A typical setup would provide a flight time of around fifteen minutes. This flight time can be extended with more batteries which are expensive and heavy.

Glow

Glow engines use a high-octane fuel. These systems can be set up to give long flight times and the engine itself is simple, cheap, and has a high power to weight ratio. However the fuel for these engines is highly flammable and very expensive. The current cost for “Nitro”, the fuel used in these engines, is around 35 dollars a gallon. These engines are better suited for smaller scale aircrafts. Also due to the nature of how these engines works, starting these engines on an airplane is rather dangerous. It requires someone to reach over the cylinder head and remove the glow plug igniter, which is about six inches from a fast moving propeller, which can easily remove a finger.

Gas

Gas powered RC airplanes use a conventional 2 stroke motor very similar to the ones found in small gas power yard equipment. This style of motor is simple and very reliable. These engines do require some care and maintenance in the setup and operation. It uses 93-octane gasoline from any fuel pump with a 30:1 2-cycle oil mixture, which is relatively cheaper than Nitro. The initial cost of the engine is more expensive than a glow engine, however the long term cost is cheaper than glow.

2.1.3 Basics of Flying

Remote Control flying is a unique and challenging experience. The entire process can be broken down into four major steps. These steps are; understanding flight mechanics, flying in an RC simulator, flying using a buddy box system then flying solo.

An airplane in flight has six degrees of freedom, x-axis, y-axis ,and z-axis position, roll, pitch, and yaw. These variables are controlled on the plane by the control surfaces. The elevator will affect the pitch of the plane which is used to help make the airplane climb. The roll of the plane is controlled primarily in the ailerons of the airplane. This will help the airplane bank and turn. A combination of the ailerons and elevator will allow the airplane to make a successful banked turned. Finally that yaw of the plane is controlled with the rudder, and is primarily used only to get the airplane pointing in a specific direction when there may be a crosswind on a landing attempt. Finally, the last variable of an airplanes controls is the throttle. Throttle does control the speed of the airplane, but it is actually used to determine the altitude of the plane. When you increase your speed on a plane, you will be passing more wind over the wings, thus causing more lift and gaining altitude. Being able to properly control the throttle is a very important step in flying and it is most critical in takeoffs and landings. It takes many hours of practice and numerous flights to be able to fully master all these variable for a successful flight.

Using an RC simulator is a great place to start, or to hone in and improve your skills even as an experienced pilot. The greatest benefit to using a simulator is that if you crash you can simply restart the program and try again without all the time and money that it takes to have to rebuild or replace your plane. However, you should still take the simulator seriously because you should not develop bad habits in the early stages of learning to fly.

One of the biggest challenges of learning to fly RC is the orientation. When flying the plane, if you change the directions from going away from you to toward you, left and right become reversed. This is the biggest challenge even sometimes for experienced pilots. The trick to overcoming this is not to think of it as it needs to go right or left but more of opposite of what it did. In practice try a slight course change and correct as needed. Flying is more of an action reaction experience, and that is the best and simplest way to think of it.

After you have spent several hours on a simulator, the next logical step is to go to a buddy box system. This allows you to fly your plane with and experienced pilot to help prevent crashes. This system works by using a cable to link two transmitters together that will allow the experience pilot to flip a switch and then have full control of the plane. There is also another great benefit to this system because you can get valuable feedback from the pilot of what you did wrong and how to prevent it from happening again. It is also recommend that even after doing some flying with the buddy box system, to go back into the RC simulator to try some maneuvers and practice landings.

2.2 Platform Execution

After gaining an understanding of the problem the team entered the execution phase of the project. During this phase the planes were designed, built, and tested.

2.2.1 Drafting a Safety Manual

There were many concerns with having a project based around large-scale model airplanes. So before construction started a safety manual was drafted to address these concerns. The safety manual accurately spelled out the different roles people would need to play during testing and how and where testing would occur. These concerns included, but were not limited to:

1). Large engines with very large propellers spinning in excess of 8000 rpm
2). Loss of flight systems during a flight which could lead to a crash
3). Transportation of fuel to the testing site
4). Accidental damage to the airframes during storage and transport

The manual needed to include procedures to avert accidents, such as procedures for how to handle the loss of a flight system during flight. The tools from organizations like the Academy of Model Aviation (AMA) and the FAA were used to help write the manual. The help of small engine experts and large scale airplane modelers were used to ensure that our safety manual covered all of the necessary procedures to help avert accidents. Once the manual was completed, it was brought to the WPI safety coordinator and liability personnel for approval. All members of the team and observers were required to sign this before the first flight.

2.2.2 Goose Construction

After doing extensive research on airplane design and construction the team first decided to try to design and build our own airplane from scratch to determine how feasible this was as an option. First a CAD model of the design was created using the recommended ratios, and NACA 2412 airfoil for the wings. This plane was named Goose. Goose had a 6 ft wing span. Several different design phases were completed before settling on the final CAD model displayed in Figure 13.

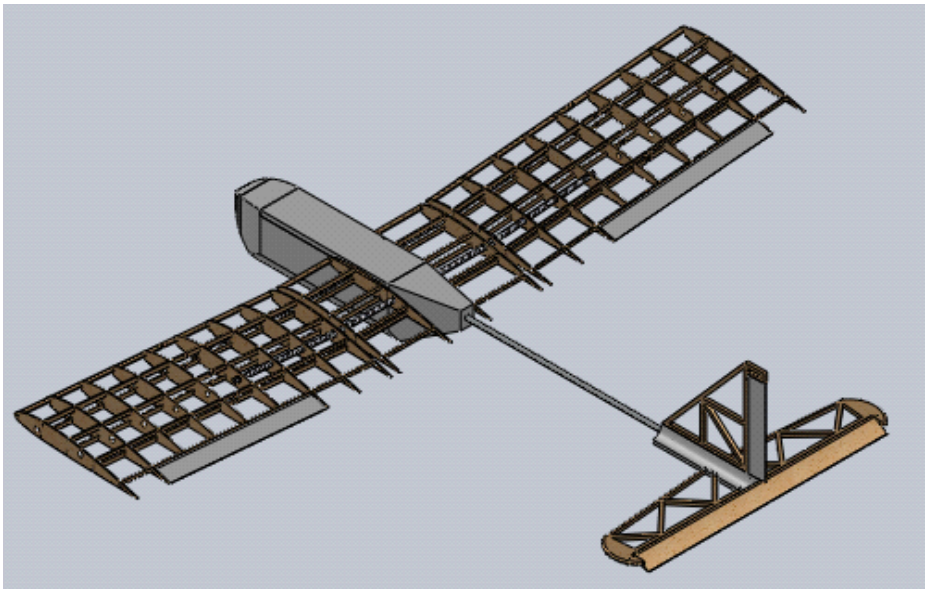


Figure 13: CAD of Goose's design before construction was started. This CAD image doesn't include the MonoKote that will cover the wings and tail and fill in the gaps.

The body of Goose was made by carving blue insulation foam with a homemade hot wire. Covering the body with fiberglass for extra support was tested, however it was too heavy for the plane, and ended up only being used around the wings attachment points. The wings were made out of balsa wood that we cut on the laser cutter to our specifications. An example of a template used in the laser cutter is shown in Figure 14.

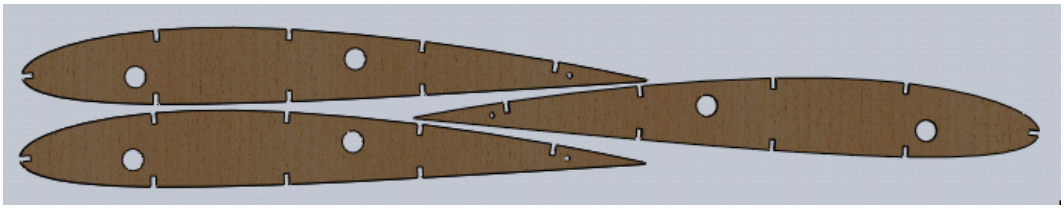


Figure 14: Laser Cutter template used to cut out the airfoil in the wings.

The boom to the tail is an aluminum tube which is screwed into the tail. The tail is a combination of balsa wood and foam. Since this plane had only a 6 ft wingspan and didn't need to have a long flight time, a brushless DC motor was purchased to power it. After Goose was fully assembled, it was determined that it was slightly tail heavy and needed additional counterweights in the nose to balance correctly for flight.

Figure 15: Fully assembled Goose, ready for flight

Designing and building Goose was a 3 month learning experience. In the end the team decided that buying partially assembled RC airframes would be a better decision for this project. The scope of the project was on creating UAV's capable of preforming search and rescue missions not on building airplanes.

2.2.3 Parts Selection

With the background experience from completing goose, the team needed to choose appropriate hardware for our drones. This next section will discuss our selection process for the necessary components.

Airframe

Our airframe selection was one of the most important decisions that the team made. The airframe determines the size and type of engine that can be used. It also determines the max weight and volume of the payload. This payload includes the fuel, all of our electronics, cameras, and communication hardware. The type of airframe chosen had an effect on its flight speed, maneuverability and sensitivity to commands. All of these factors were considered in airframe selection.

For the first drone, the team selected the Senior Telemaster. The Senior Telemaster has a 7.83 ft wingspan, and 9.24 square feet of lifting area. It is a trainer plane and is notoriously easy to fly for beginner pilots. It also has a 10 pound payload capacity, which fits our expected payload. Several other college teams are working on UAV projects using this airframe(University of Alberta, 2011). This plane was named Robin.

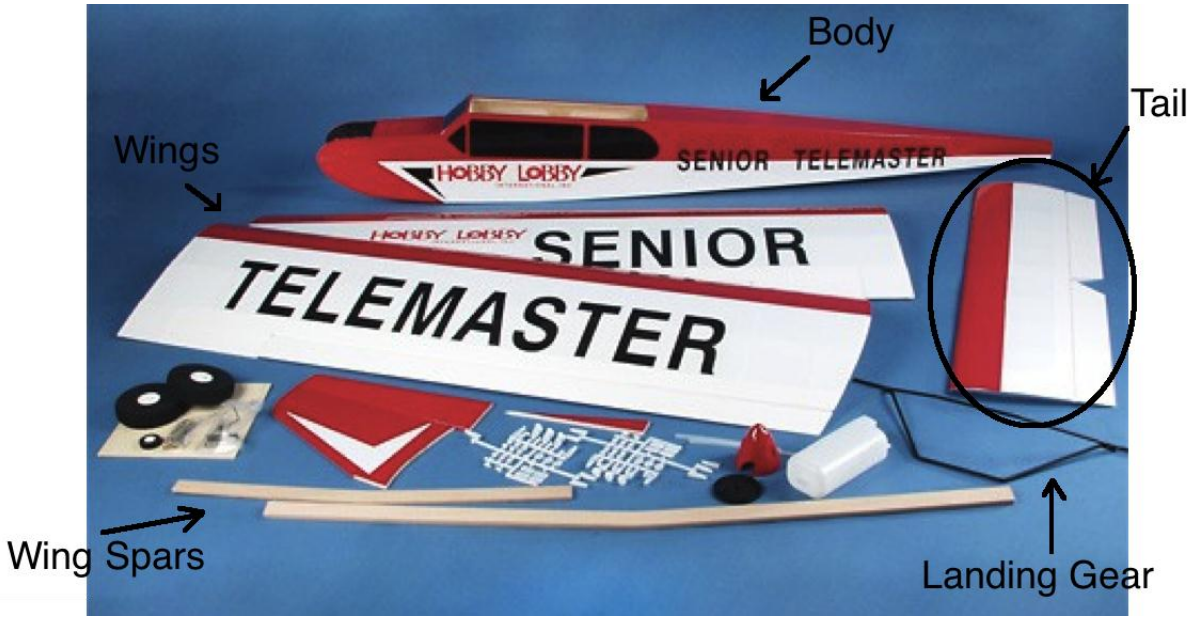


Figure 16: Unassembled Senior Telemaster Parts as they are shipped

For the mothership a larger payload was needed because the mother ship needs to fit the large USRP2 radio and its supporting hardware for the base station downlink. A similar airframe to the Telemaster was desired so that it would as easy to fly and so that many of the configurations from the Telemaster could transfer over to the new airframes. The team selected the 8.8 ft wing span airframe, the Skyline Champ, with 11.5 square feet of lifting area. Two of these frames were purchased, the blue one was named Jay and the yellow one was named Duck. Like the Telemaster, this airframe is also a trainer style and easy to fly.

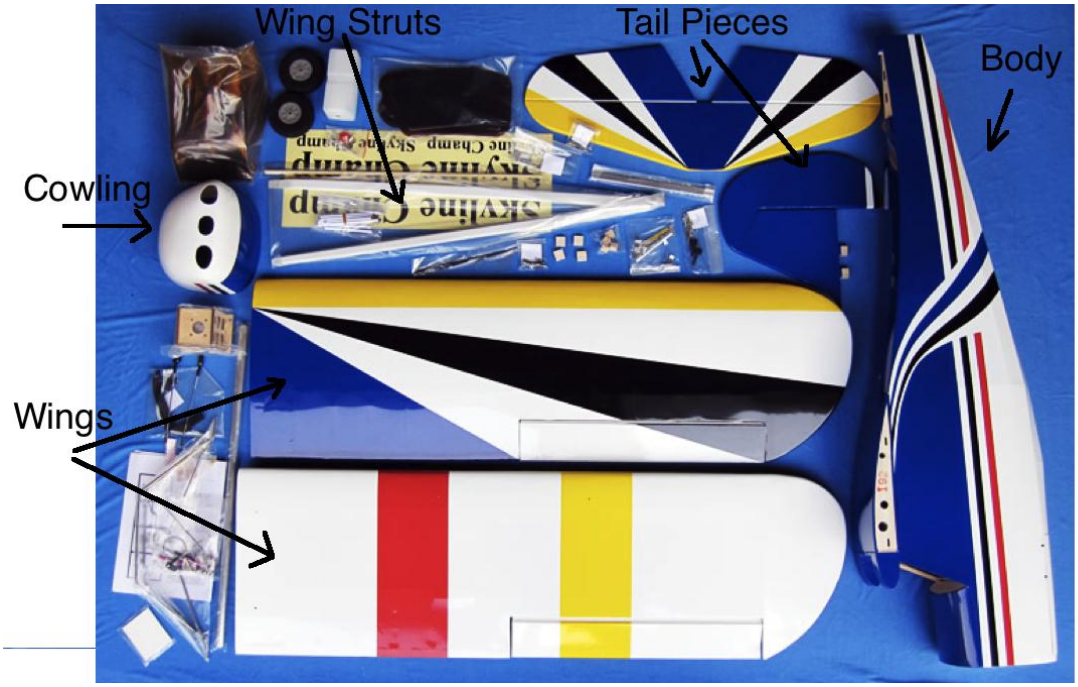


Figure 17: Unassembled Skyline Champ Parts as they are shipped

In general larger airplanes are easier to fly then smaller ones because they are more stable in the air due to a higher moment of inertia. This is to our advantage when flying under autopilot and doing video surveillance.

Propulsion

After considering all of the engines and motors available for our two airframes the team decided on gas engines. This was chosen because of the care and maintenance for the engines is not too difficult, and the fuel is readily available. The long term cost for the gas engine system compared to the nitro was less. An electric motor for the class of airframe would require a very large, heavy and expensive battery to meet the requirements. The cost of the motor alone would be around 170 dollars plus 290 dollars for the batteries needed to power the motor for our targeted flight of an hour. The power source for an electric system is more efficient because it would take up less space, but fuel is more energy dense(Golnik 2003) (Barnard Microsystems Limited 2012) than the batteries and significantly cheaper.

Figure 18: Stock RCG 20cc Gas Engine out of the box

The RCG 20cc Gas Engine (2.2HP)was chosen from HobbyKing.com to power the Senior Telemaster based on recommendations from RC experts.

The RCG 30cc Gas Engine (3.9HP) from HobbyKing.com was chosen to power the Skyline Champ. This engine fell within the recommended range of engines for the airframe by the manufactures. An engine on the lower end of the range was chosen because the high performance a larger engine would provide was not needed.

Gas tank

A larger gas tank allows a longer flight time. However if the tank is too large for the airplane and not placed properly the pressure differential can drop the suction for the engine during aggressive maneuvers and stop fuel flow thus stalling the engine. Because of these factors, a 32 oz tank was chosen for the Telemaster, which gave the airframe a nice balance of both size and functionality. A 40 oz tank was chosen for the Skyline champs to the airframe comparable flight time with the Telemaster.

Propeller

Propellers have two important dimensions that affect their performance, the diameter and pitch. The pitch affects the top speed of the airplane. It is defined as the distance the propeller would move in one revolution if it were moving through a soft material, like a screw through wood. A lower pitch will force the engine to reach max rpm at a lower speed. A larger pitch has a higher top speed but slower acceleration. The diameter of the propeller is determined by the desired rpm and power of the engine. Since a slow moving airplane with a lot of power was desired, a large propeller with a small pitch was selected. On the Telemaster a 16 inch diameter with a 4 inch pitch (16-4) was ideal. On the Skyline Champs a 18 inch diameter propeller with a 6 inch pitch (18-6) was ideal.

2.2.4 Robin Air Frame Modification

Before it was flight ready a few modifications were made. The first modification was adding 2 degrees of trust to the right to the preexisting engine mount. To balance all of the internal components inside the fuselage, a CAD model of the Telemaster fuselage was created, and recreated all of the

components in CAD as well. Two different layouts for the interior were created; one with all of the components and one with only the components necessary for autopilot fight.

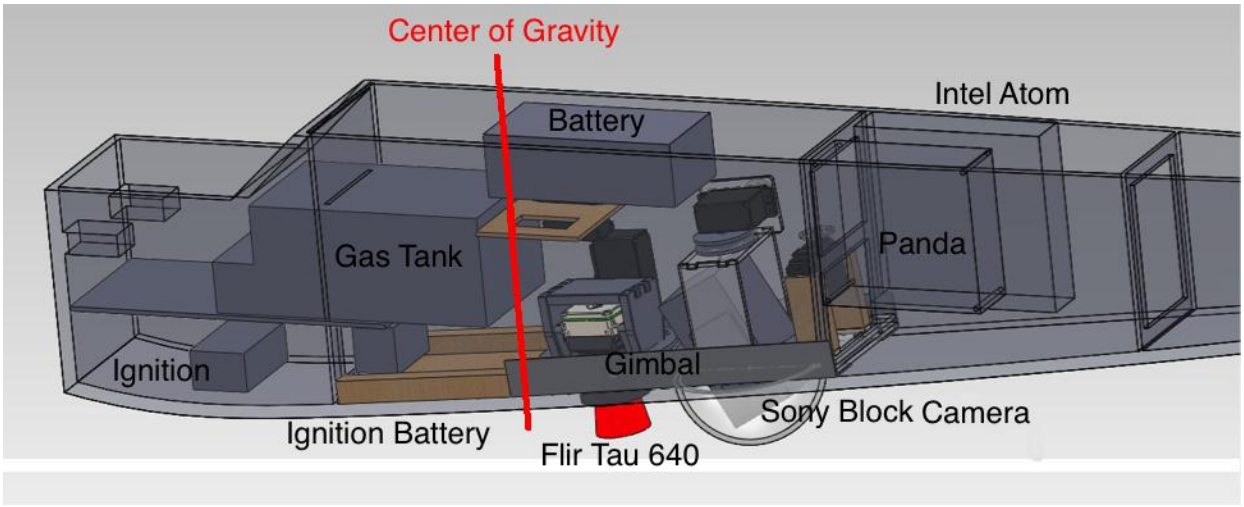


Figure 19 Internal layout with all of the components from all the teams

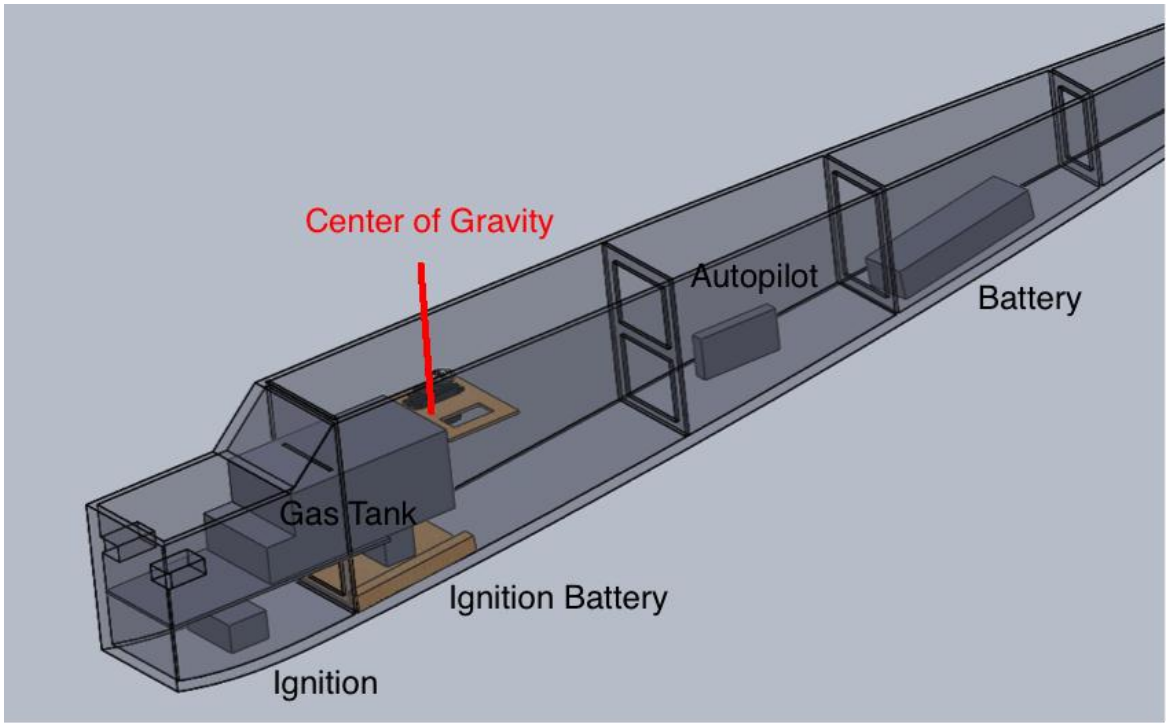


Figure 20 Internal layout with only the components necessary for flight

While examining our CAD drawing, it was determined that the rear compartments in the fuselage were needed for space. Hatches were installed utilizing pieces of wood and a hinge, which allowed access to components in the rear of the plane with ease as seen in Figure 21.

Figure 21: Robin with back hatches installed

The next major change made was the location of all the internal servos. The plane was designed to have all of the servos in the middle of the plane, under the wing, however that area needed to be cleared. Also the way the servos originally controlled the rear control surface was unreliable. It was using long pushrods that could flex slightly, undesirably allowing the elevator to shift out of position under load. The problem was solved by shortening the push rods and moving the servos to the back of the plane, utilizing the new hatches as seen in Figure 22.

Figure 22 Robin with the servos moved to the back of the tail

The next major modification made was installing the gas tank. Initially a gas tank was installed as far forward and up against the front firewall of the airframe. A floor in the tank compartment was installed and walled it off from the rest of the plane interior compartment. It was made fuel proof by painting the compartment with oil based paint and sealed the cracks with hot glue as seen in Figure 23.

Figure 23: Gas tank in its original position all the way forward

The rest of the modifications where made based on the results gathered in our flights. The flights are discussed in detail the testing section.

2.2.5 Camera Gimbal

A camera gimbal capable of containing two different cameras, the Sony block camera (color), and the Flir Tau 640 (thermal infrared) was designed. The gimbal serves two different purposes, to maximize the cameras view and reduce vibrations from the engine and propeller to the camera. Each camera had to look down through their own dome instead of sharing one, which meant they each had to be on their own gimbal set up.

Figure 24 Sony color block camera that we will put in the planes

Motion from the servos is transferred to the gimbal by linkages. Linkages were chosen over gears, pulleys, and strings because of the vibrations in the system caused by the engine. The vibrations would cause the gears to quickly get worn out and increase uncertainty in its position. Gears would also transfer and magnify all of the vibrations going through them. Pulleys are better at not

transferring and magnifying vibrations than gears, however the vibrations will cause them to slip and the actual location will be off from the expected location. Over the duration of the flight, it could completely shift our range of motion away from the ground. A string in a pull- pull set up with the servo was considered, though the string will not help absorb the vibrations and over time will be worn out and begin stretching. Since the linkage system is made from spring steel, which deflects under the faster vibrations while remains solid when responding to the servo, it successfully absorbs the vibrations without compromising its functionality. However the pan rotational motion was done with a pulley system because it was not going to be largely effected by the vibrations. The pan motion was not supporting a rotating mass that contained inertia. It was also mounted perpendicular to the majority of the vibrations in the system.

This entire structure is supported on vibration dampening rubber that isolates it from most of the airplanes vibrations. First the elastic properties of different materials typically used in vibration dampening mounts like rubber and silicon were researched. The team was also able to find several data sheet that showed the frequency ranges for different silicon dampeners. Based on the engine data sheet the engine would be running between 30 and 40 Hz while flying. The graphs in Figures 25, 26, and 27 depict the material properties of three different silicon with different Shore A hardness indexes. Shore A is standard for measuring the hardness of different soft solids. The optimal load for each of these materials was based on their thickness.

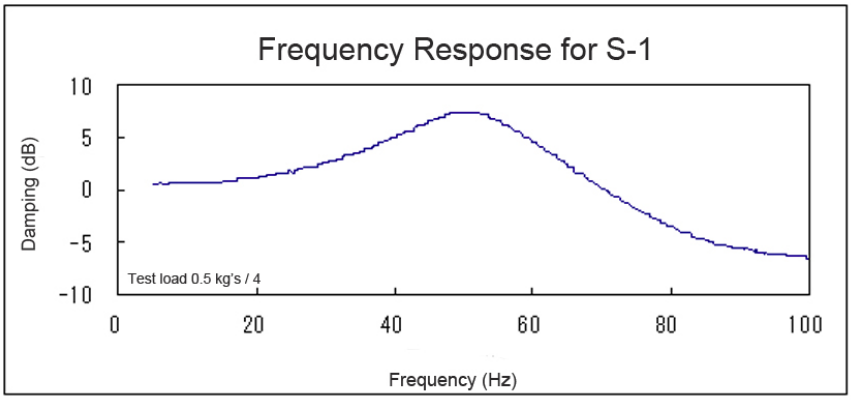


Figure 25: Frequency responded for a silicon with a 6 on the Shore A hardness scale and optimal load of 0.2~0.75 (kg/ 4 legs)

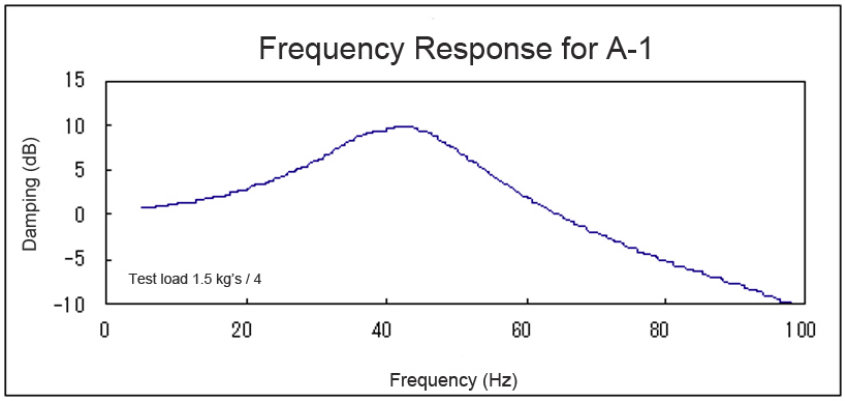


Figure 26: Frequency responded for a silicon with a 10 on the Shore A hardness scale and optimal load of 0.5~2.5 (kg/ 4 legs)

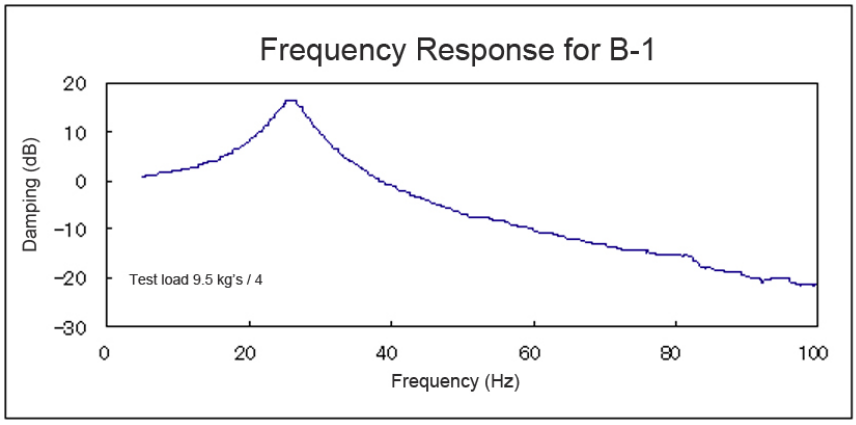


Figure 27: Frequency responded for a silicon with a 14 on the Shore A hardness scale and optimal load of 4~15(kg/ 4 legs)

Different materials protect against sheer and compressive vibrations differently. Based on this information a material with a Shore A hardness index of 12, whose optimal load was about .5 kg per connection point was required. This loading was based on the CAD model and the weight of the components it needed to support. This led the team to purchase a 1/8 inch thick silicon based foam from McMaster Carr.

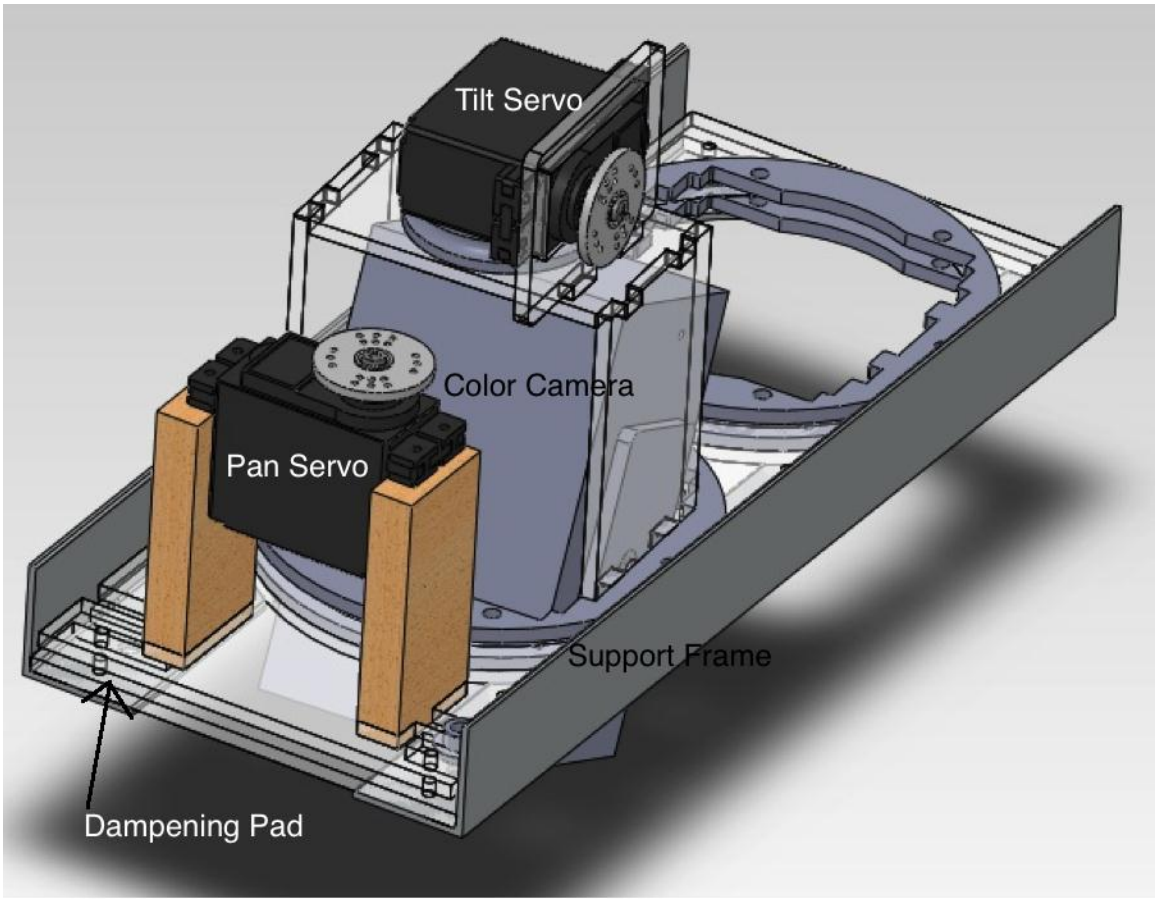


Figure 28 CAD of Camera Gimbal with Sony camera

However because of a lack of funding we were unable to acquire a Flir camera. However the option was left available for future teams to add the camera to the plane. Until then the battery was strategically placed slightly forward to compensate for its absence.

2.2.5 Jay Air Frame Modification

With the previous experience from Robin, the team was able to make better decisions when constructing and modifying Jay. For instance, when installing the fuel tank a decision was made to install it a quarter of the chord from the leading edge of the wing. This ensured that the fuel tank will not interfere much with the center of gravity (CG) as the amount of fuel in the tank decreases during the flight. This allowed balancing the plane and keeping a constant CG simpler.

In order to install the fuel tank, a fuel tight box was made out of 1/8 inch plywood that was slightly larger than the tank as instructed by the manufacturer for a pressurized fuel tank. This box was installed into the plane, but to do this part of the rib structure had to be cut away. Reinforcements were made to the areas cut away with 3/8-inch square dowels to help support the tank and the airframe as seen in Figures 29 and 30.

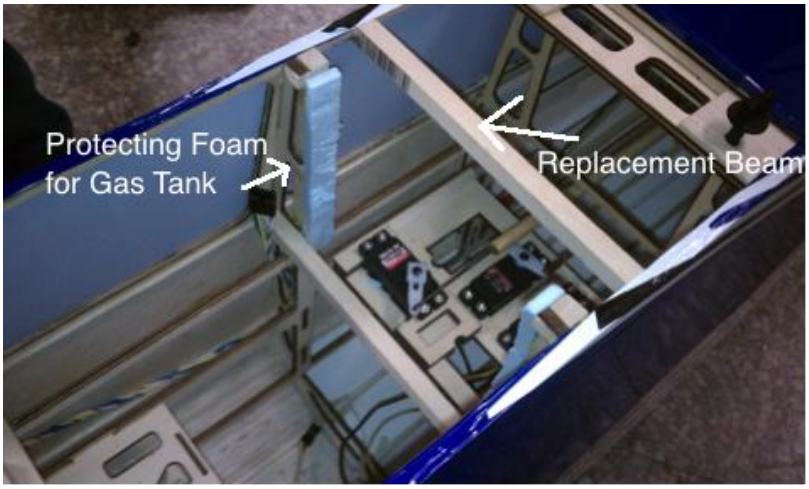


Figure 29: Modifications to fit the gas tank chamber

Figure 30: The fuel tank chamber in place in the body

The next modification was the installation of the quick disconnect valves for easy refueling of the plane. After our experience with Robin, it was found that there was a considerable amount of force needed to use the valves so appropriate measures for the installation were applied. A piece of 1/8 inch plywood was glued behind the outer sheet of the body of the airframe to reinforce the area where the valves were installed as seen in Figure 31.

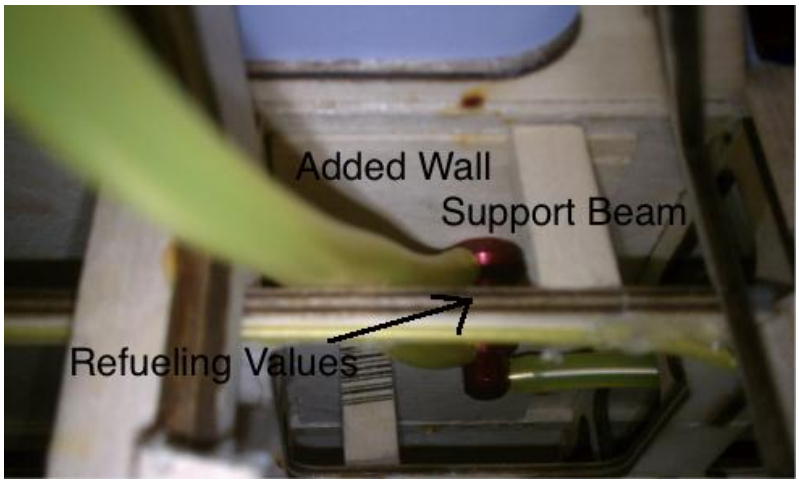


Figure 31: The quick connect values from inside the body of the plane, with reinforcing plate

From flying Robin the engine had to have some right and down thrust built in to the mounting of the engine for straight predictable flight. Jay came with a wooden box intended for the electric motor that had these offsets built in. These standard offsets were 3 degrees down and 3 degrees to the right. Since an electric motor was not used, these offsets needed to be recreated for the gas engine. When mounting the engine, washers were placed behind the mounting points to duplicate this offset. The motor was installed on a vibration dampening plate to help protect the electronics from the engine vibrations. This plate was mounted to the airframe using four ¼-20 bolt through rubber grommets as seen in Figure 32.

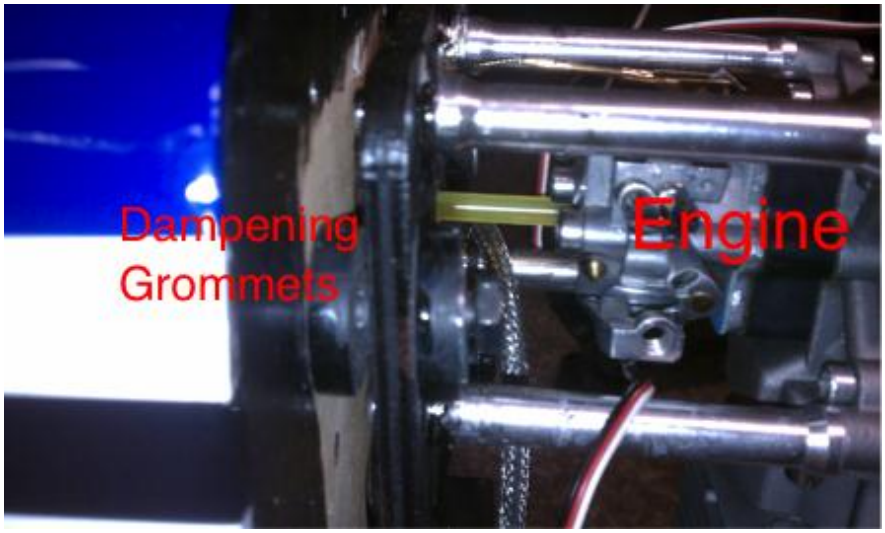


Figure 32: The rubber grommets attaching the engine to the fuselage

Finally for the throttle linkage a flexible cable was used after the frustrating attempts to use a solid linkage in Robin. Unlike Robin, the return spring was kept on the carburetor to allow better response in decreasing the throttle with the cable system.

2.2.6 Duck Air Frame Modification

Because of the setbacks from Robin, the team made the executive decision to stop Ducks construction half way and shelf him. The plane will have to be completed by future teams before Duck can join Jay and Robin in the air. The progress on Duck is summarized in the list below.

- Engine broken in
- Propeller modified for engine
- Landing gear attached
- Tail attached
- All control surfaces attached
- Servos for the ailerons, elevator, and rudder mounted
- Control linkages for the ailerons, and rudder are completed
- Rear landing wheel is mounted

Figure 33: Yellow Skyline Champ Duck, partially assembled

For Duck to be completed more items need to be purchased. For RC flight Duck needs new wheels, a gas tank, an engine dampening plate, flexible linkage for the throttle, a receiver, and two quick disconnect valves. For autonomous flight it would need a YAPA2 autopilot and GPS. All of the other sensors and supplies have already been purchased. For Duck to be equipped to satisfy the goals of Project WiND and have coordinated search flight with the other planes it will need the higher level boards, an image processing board, power board, Wi-Fi module, and a camera mount designed and installed.

2.2.7 Power Board Design

As the plane came together and the other teams got their ideas together of what they will need for their projects, the need for a power management system became more apparent. This power system needs to be able to power all the systems on the plane as well as to be able to turn off unnecessary systems when power starts to get low. The list of priority items can be found in Table 2.

The purpose of having this priority list is to determine what items can be turned off without losing functionality of the flight of the plane for as long as possible in low power situations.
For the independent systems it was determined that it would be best if each had its own controllable regulator. The design used an LM5118 which is a wide voltage range buck-boost controller. This particular controller can take an input from 3 Volts to 75 volts and output it from 5 to 15 volts. As for the current, this devise is only a controller for the switching of the MOSFETs for the buck boost converter so we just specified ones that can handle up to 10A.

Table 2: Power board specifications

System	Voltage	Current	Allowed Power	Priority level*
FPGA Power	5 V	5 A	25 W	4
Camera	6 V	.5 A	3 W	4
SDR Amplifier and Computer	5 V	4 A	20 W	3
SDR	6 V	3 A	18 W	3
WIFI	15 V	.8 A	12 W	2
Panda	5 V	1 A	5 W	2
Autopilot System	8 V	.5 A	4 W	1
Servos	5 V	3 A	15 W	1
Total			102 W	

***priority goes from highest being 1 down to 4 the lowest**

Each regulator circuit is using the same fundamental design with the only difference being different values in a few of the resistors or capacitors in the supporting circuitry. For instance the ratio of the resistors R8 and R9 will determine the output voltage by this equation, $\frac{R8}{R9} = \frac{Vout}{1.23} - 1$. The calculation and description for maximum output current is a little more involved and is best described in the data sheet for the LM5118 which can be found in the Appendix I.

With the setup the regulators are going to be useed around their most efficient point. For this regulator, the efficiency is dependent on the input voltage and the highest efficiency is when Vin is around 18V which is near the voltage of both a 4 cell (14.8V) and a 5 cell (18.5) Lithium Polymer battery.

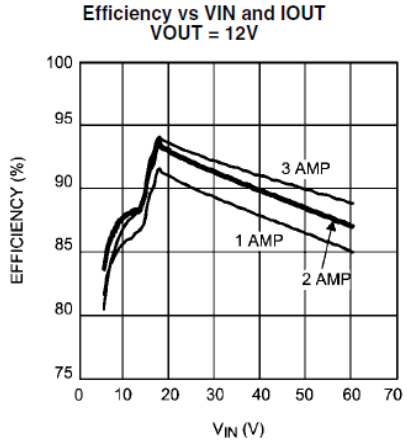


Figure 34: LM5118 efficiency curve

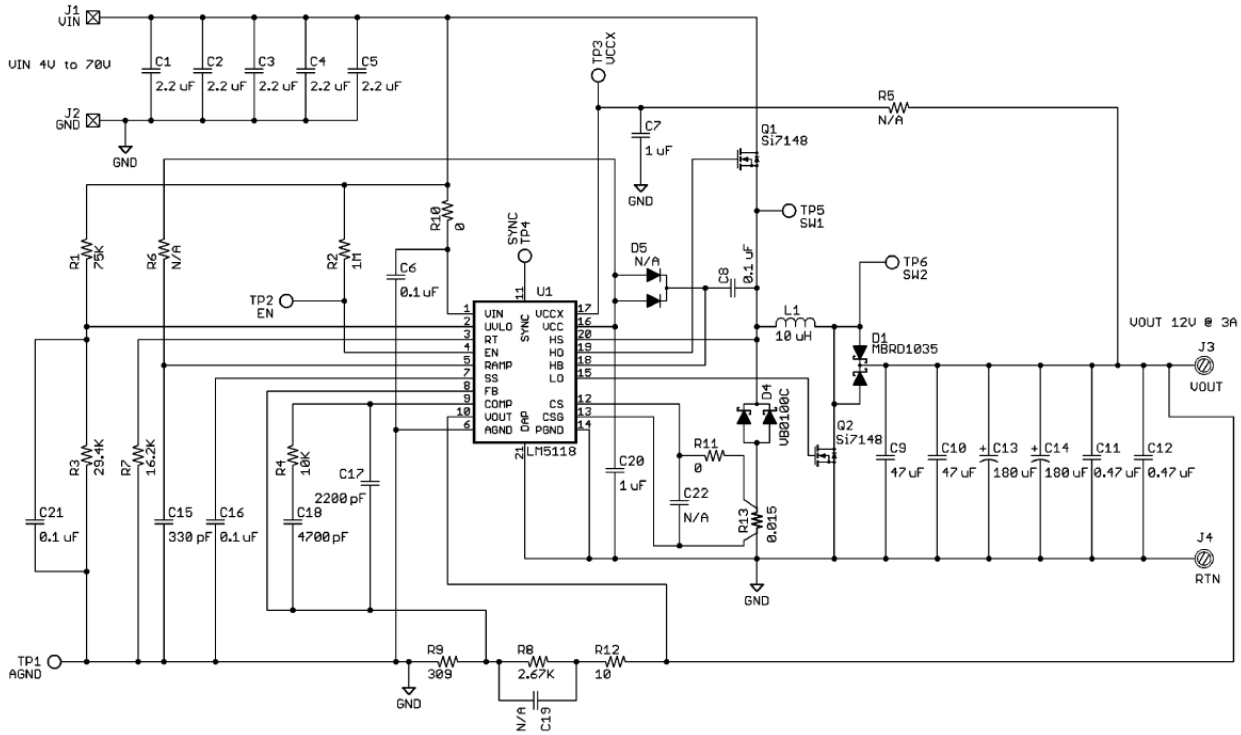


Figure 35: Standard configuration for buck-boost regulator

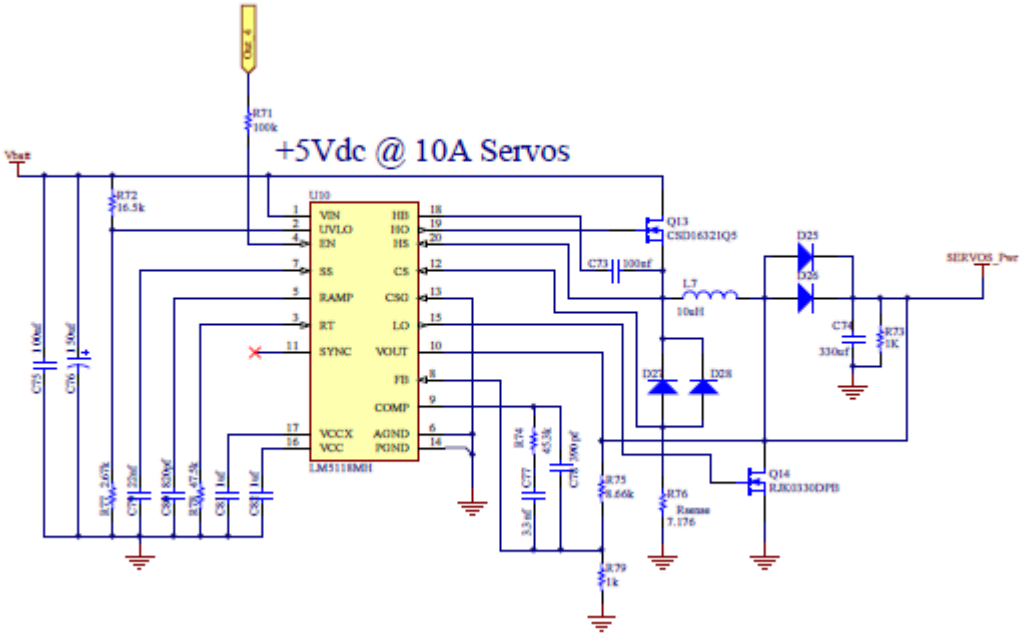


Figure 36: Configuration used in our schematic

For the control logic for this system an MSP430 was used. A simple controller was needed to monitor the current battery voltage, and to control the enable lines on all the regulators. Using a microprocessor for this simple task is unnecessary however, the decision was made because it would leave room for expansion for future project groups, and allow the system to be easily adapted if the battery type is changed.

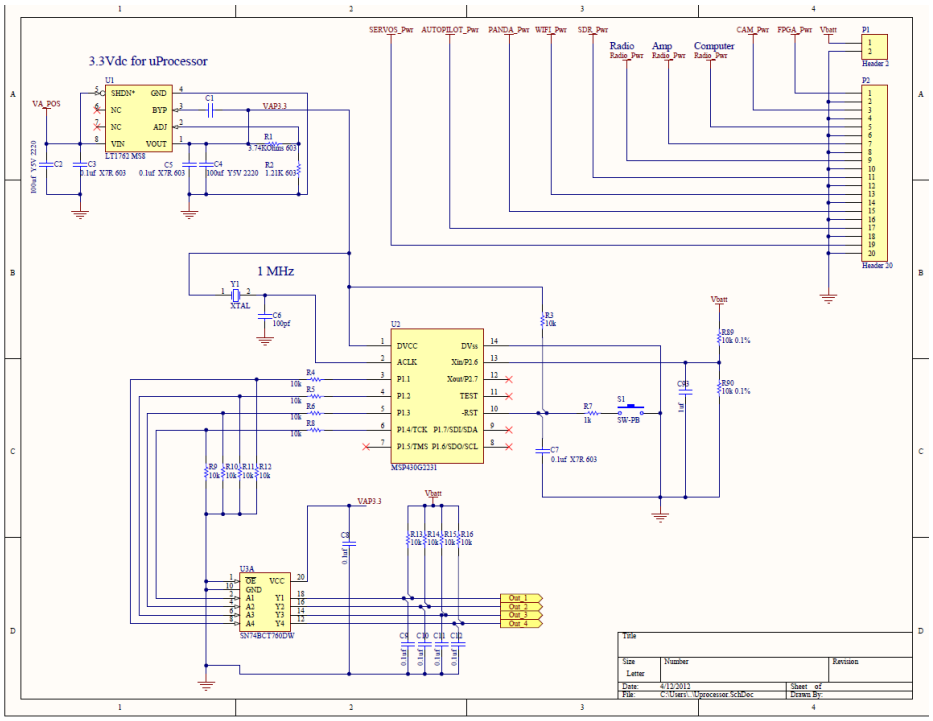


Figure 37: Controller Circuit Diagram

The battery needed for this system will have to be able to produce a continuous 102 Watts for the system to fully operate. This was calculated using the worst case conditions, for example, stall condition for servos, then round up to the nearest tenth of an amp just as a precaution. As for battery life, it will need to last for at least an hour based on our expected flight time. To meet the standards and to try to keep it as light as possible the team is recommending the use of a lithium polymer (LiPo) battery. LiPo batteries are commonly used in the RC as the main power source for planes. LiPo batteries have a high energy density and are moderately safe to handle and charge unlike lithium Ion batteries (Barnard Microsystems Limited 2012).

This system has yet to be fully developed due to time and budget constraints on the entire project, but the design of this board is finalized and ready for the next steps in production.

Testing

The rest of the modifications made to the airframes were based on the results gathered during the test flights. Figure 38 shows some still shots from one of Robin's takeoffs. Both of these were stable takeoffs, short and straight up without any power steering during acceleration or hard pulls once they were in the air.

Figure 38: Images from the video of Robin's Take off out at Tanner Hiller.

Figure 39: Images from the video of Jay's take off at Tanner Hiller

Testing occurred on seven different occasions. Each of these trips had a different goal with different lessons. Table 3 summarizes the results from these flights, and the resulting changes to the airframes.

Table 3: Test day summary results based on outcomes

Date	Goal	Outcomes	Resulting Changes
Nov 13	Robin RC Flight	-During Aggressive climbs the plane pulled to the right -Engine had a non-linear response to throttle control. -Engine stalled in final flight	-Altered engine mount thrust angles -Changed rudder servo to faster one -Moved the gas tank back 4 inches -Moved the throttle servo forward -Pressurized the gas tank with muffler -Fix rudder and elevator horns
Feb 2	Robin RC Flight	-Successful GPS downlink -Nipple of muffler melted off -Wings shifted and sheared off wing struts connection	-Wing strut replaced with stronger alternatives -Reduced the throw of the control surfaces -Added Aluminum Frame -Added Hard Mount points on wings -Mounted the IR sensors
Feb 9	RC Flight Through decoder Board	-Calibrated thermo sensors on plane -Sideways landing broke wheel	-Re shrunk the monokote -Replaced the broken wheel
Feb 15	Auto 1	-30 cc engines broken in -Found we couldn't taxi strait with new wheels -Plane pulled to the right during stable flight -Lost the down link before going into Auto 1	-Fixed the torqued wing -Fixed the orientation of the wheels -Added New decoder board
Feb 18	Auto 2	-New decoder board interprets signals differently -Caught a cross wind and crashed	-Fixed all crash damage -Installed the dome -Calibrated new decoder board -Replaced glue on tail
Apr 7	Jay Flight Auto 1/2	- Too windy, taxi test only	-NA
Apr 11	Jay Flight Auto 1/2	-Jay flew, slight warp in right wing discovered -Only Half of Robin returned	- To be continued

After the first flight a redesigned of the internal layout of Robin was completed. With the new layout, the gas tank was moved aft by about 4 inches. This helped to stabilize the location of the center of gravity, while still keeping it in a good location with respect to the engine. The engine in the new location can be seen in Figure 40.

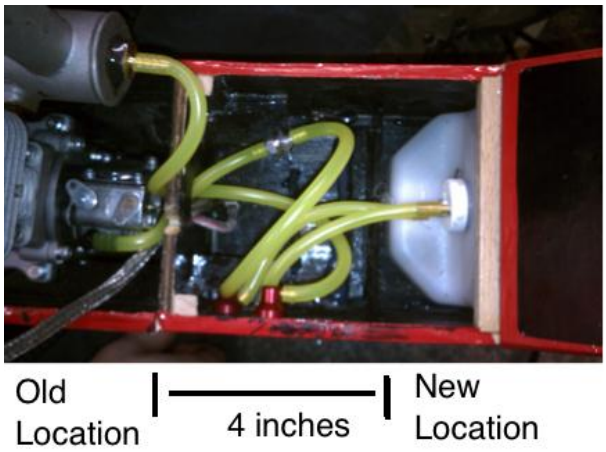


Figure 40 Gas tank in its final position moved 4 inches back

Moving the gas tank aft allowed the throttle servo to be relocated closer to the engine. For the first flight, the throttle servo was mounted behind the tank and ran a thin push rood above the tank to the engine. This didn't work well because it didn't have a direct path to the engine so some bends were added in the wire, which caused it to flex giving it some problems. With the servo installed in between the engine and the gas tank, it had a more direct route and made debugging the engine problems easier.

For the first flight, the electronic ignition system was installed and secured in the forward compartment under the gas tank, and the ignition battery was mounted behind the gas tank with the cables running under the tank to the ignition coil. After the gas tank was moved, the cable was not long enough to reach the ignition coil. This required an access hatch to be installed to the lower compartment. The hatch allowed easy access to this compartment, which is where the ignition battery and the ignition coil are now stored for flight. A cross section view of the this compartment and the floor which the gas tank rests on can be seen in Figure 41.

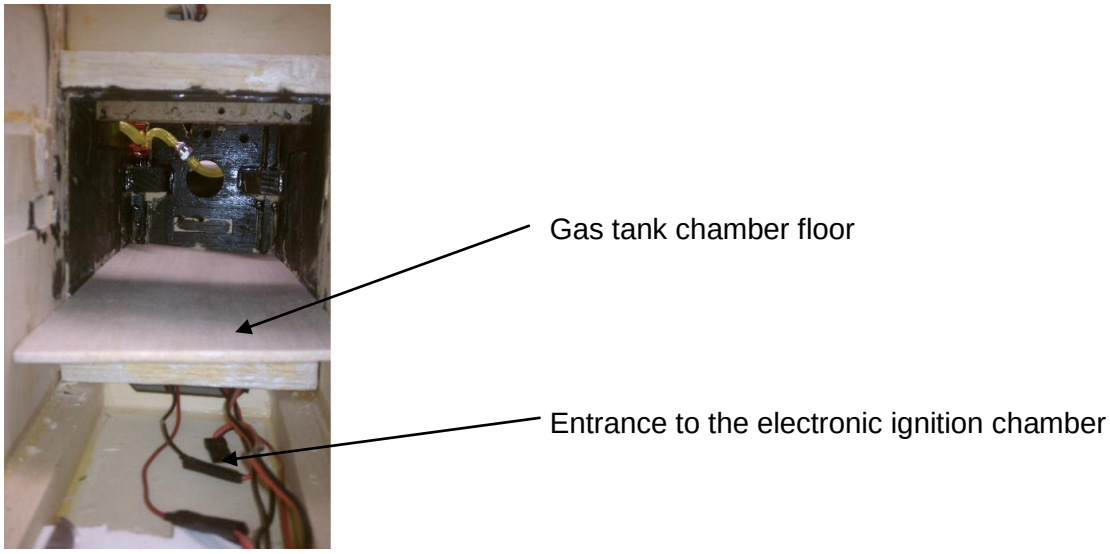


Figure 41 in side of Robin showing the different compartments

The most significant modification to the airframe was the wings attachment method. This change was deemed necessary after the airframe sustained damage during one of its earlier flights. On that flight, the wings shifted breaking one of the attachment points on the wing struts. The wing attachment was always a concern especially since the team needed to load up the plane with a lot more weight than the plane was intended for. To fix this problem, the team made the wings attach in a similar way to the Skyline Champ. Pegs were installed on the leading edge of the wings and drilled holes in the rear of the wing to allow a ¼-20 screw to secure it to a piece of hardwood with receiving nuts embedded in it. The piece of hardwood was secured to an aluminum frame which was custom made for the main compartment, as shown in Figure 42. This frame reinforces the body and helps spread the load throughout the airframe. This frame also supported the pan tilt camera gimbal that was designed. This frame was in the original design but we didn't have a chance to add it until after the second test.

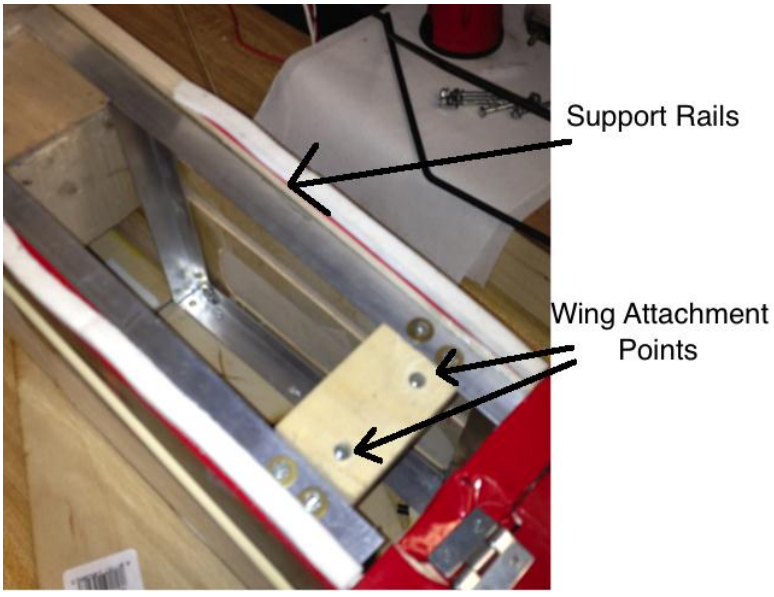


Figure 42 Aluminum support frame inside of Robin, with the wooden mounting plate for the wings

The February 18th crash sheared the landing gear off of the fuselage and took most of the floor with it, as well as the wing strut attachment points. Pictures of the plane after the flight can be seen in Figures 43.

Figure 43: What remains of the bottom of Robin's fuselage after its February 18th crash. Notice how it is missing its landing gear and wing strut attachment points. The Aluminum frame prevented further damage.

A picture of Robin after the repairs can be seen in Figure 44. All of the MonoKote on the front of the plane got ripped up during crash, and therefore had to be replaced. Thankfully, the engine shaft was undamaged.

Figure 44: Robin after its repairs from the February 18th crash.

On April 11, Robin went out for its final test. As part of our integration, a camera was mounted in Robin, as well as its supporting hardware to record video. Figure 45 shows its set up in the plane inside the dome. Since a camera was installed for this flight, and the it was recording, the software team requested a few fly overs to capture images of people on the ground while still under RC control before testing the autopilot.

Figure 45: Camera behind Robin's dome before its April 11th flight

At the end of these passes control of the elevator was lost and it flew into some trees. The team was able to recover the engine, all of the electronics, and three of the servos. The tail right wing and the landing gear were not found after this incident. Figure 46 shows what was left of the fuselage after the crash.

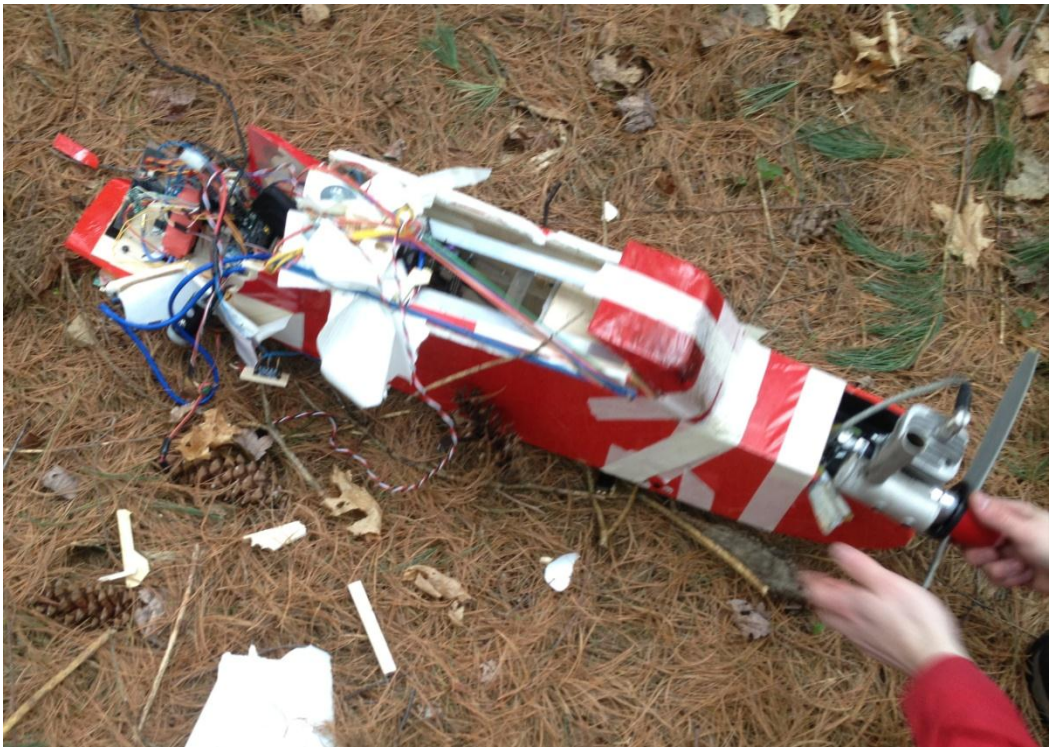


Figure 46: What was left of Robin's body after the April 11th Flight. All of the electronics survived the crash.

Robin's airframe is currently beyond repair and it will have to be replaced before any future flights.

2.4 Platform Results

This section describes how well the team met its design specifications for the platform. These results also include how well the gimbal did at dampening the vibrations to the camera.

Completing Platform Goals

The goals for a platform team at the beginning of the project where to develop three planes modified to support autopilot, camera systems, AI systems, and radio communication. Support is defined as having proven that plane we made to have the lifting power to lift them, the space to contain them, and the electrical power to operate them. Table 4 shows exactly what the platforms are prepared/planned for, by airframe. Even if the opportunity was never realized during this project to implement all of these specifications, the ground work has been laid out for future teams to utilize, particularly for integration between the teams. For example, the final Layout for balancing for all three airframes in all configurations were not completed during this project, however the layouts were complete in CAD, illustrating that the groundwork is ready to be implemented.

Table 4: Shows the different types of support that the team completed preparations for on the different airplanes

	Robin			Jay			Duck		
Prepared for:	Weight	Space	Power	Weight	Space	Power	Weight	Space	Power
RC Flight	X	X	X	X	X	X	X	X	X
Autopilot Support	X	X	X	X	X	X	X	X	X
Camera Support	X	X	X	X		X	X		X
AI System Support	X	X	X	X		X	X		X
Comms Support	X	X	X	X		X	X		X

Table 5 shows which of the following types of support demonstrated during at least one of the test runs.

Table 5: Shows the different types of support that were Implemented on the different airplanes

	Robin			Jay			Duck		
Implemented:	Weight	Space	Power	Weight	Space	Power	Weight	Space	Power
RC flight	X	X	X	X	X	X			
Autopilot Support	X	X	X						
Camera Support	X	X	X						
A1 Support	X	X	X						
Comms Support									

Flight Data from GPS

Utilizing the GPS, graphs were made of our altitude and inferred velocity during all of the flights that data was recorded. During the tests, an average speed of 40 miles per hour was achieved which falls within the design specifications. Figure 47 shows the different altitudes the plane achieved during the test based on sea level. Based on this chart it is known that the airframe reached over 1000 feet and averaged around 200 ft. Figure 48 shows the Velocities for the same flight test.

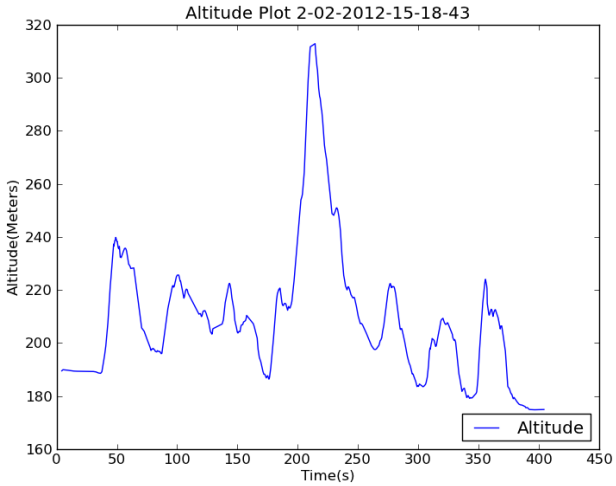


Figure 47: Altitude Plot based on the February 2nd flight.

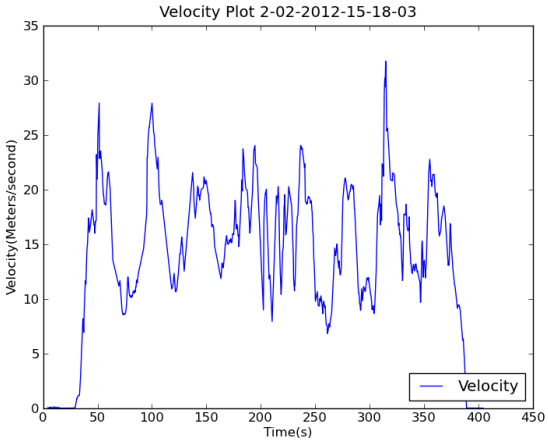


Figure 48: Velocity plot from the February 2nd flight

Gimbal Results

During the first Robin flight, a hard mounted camera was installed to the bottom of the plane and recorded footage of the flight. During the last Robin flight, a camera was added to our camera gimbal, therefore utilizing the vibration dampening pads. Its footage cut out while the plane was still on the ground, but the team was able to compare the images of the ground with the engine running for these two cameras. Figure 50 shows a screen capture of the footage from the camera hard mounted to the bottom on the first flight. Figure 49 shows a screen capture of the footage from the camera mounted in the plane with vibration dampening. Notice the distinct sine waves in the image that was hard mounted to the frame as compared to the footage with vibration dampening. Eliminating these waves in our video feed make the image processing algorithms more accurate since they have to do less filtering. In some cases the sine waves are so distinct that even after filtering them out the image would be too disfigured to accuratly run the algorithms on.

Figure 49: Camera image with the engine off, when the camera is mounted properly with vibration dampening

Figure 50: Camera images with the engine on, when the camera is simply taped to the bottom of the plane.

3. UAV Control Design

The second critical component to this UAV project is the control system. The control system is in charge of driving the control surfaces of the plane based on external sensor measurements. This section will cover the background research of existing control systems and external sensors commonly used in UAVs, the approach to integrating a control system with a UAV system, and the corresponding results.

3.1 Background

The two major components of a UAV control system is the autopilot system and the flight sensors used. The autopilot system has control over all of the control surfaces of the UAV and senses the UAV's environment based on the flight sensors used.

3.1.1 Autopilot

In order to have the UAV's fly autonomously they will need an autopilot control system. Designing an autopilot system from scratch is a complicated procedure and was highly suggested against by UAV communities and other university programs.

An autopilot system can be generalized in Figure 51.

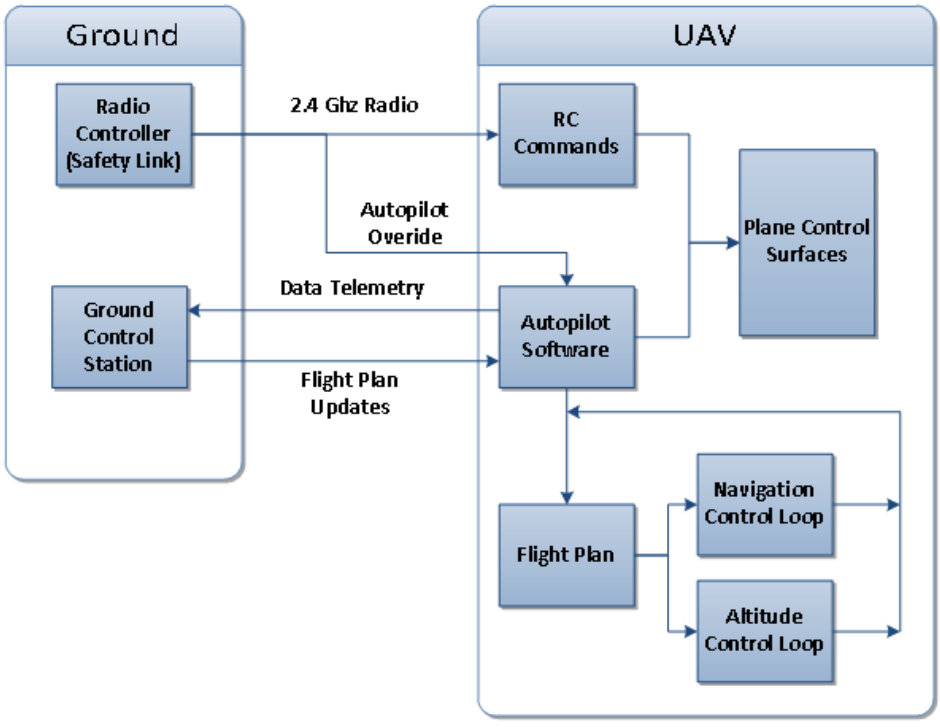


Figure 51: Autopilot Flow Chart. This chart represents the flow of how the autopilot system interfaces between the UAV in the air and the user controlled ground station.

An autopilot system consists of autopilot software, which is run on the UAV. This software is in charge of flying the UAV and relaying sensor information to a ground station. The autopilot software is preloaded with a flight plan that guides the UAV through a sequence of waypoints. Based on the UAVs current position, the altitude and navigation control loops manages the UAVs control surfaces so that it is directed to the next waypoint in the flight plan. The ground control station is a portable computer running software that receives sensor readings from the UAV and presents it to the user via a graphical interface. The ground control station provides the user the ability to update the UAVs flight plan during runtime in the event that the UAV needs to be redirected. As a safety backup, the autopilot system provides a method to interface a radio controller with the UAV, which has the ability to switch off the autopilot and provide manual control.

There are many commercial and open source UAV autopilot solutions; one commercial option and two open source options that are commonly used among professional and amateur UAV developers. These solutions include Piccolo, ArduPilot, and Paparazzi.

3.1.1.1 Piccolo

Piccolo is an autopilot system developed by students from MIT. The piccolo autopilot is an all in one system for flying a UAV and uses proprietary software designed by MIT students and graduates. The package includes the autopilot software, flight sensors, navigation sensors, and wireless communication. This product is considered successful in the field of UAVs although it costs $4,000 after academic discounts. This autopilot system is a black box system where the components are difficult to exchange with new components. Many UAV research projects utilize the Piccolo autopilot (Almeida et al, 2007) (Garcia et al, 2007) (Almeida et al, 2006).

Figure 52: Piccolo autopilot system. Piccolo is an all in one solution for autonomous flight provided by MIT. http://www.cloudcaptech.com/piccolo_II.shtm

3.1.1.2 ArduPilot

ArduPilot is an open source autopilot project similar to Paparazzi that runs on modified arduino megas. Arduino megas are readily available development microprocessor boards. It is known for its simplicity and easy to use ground station software. The ground station software expects sensors only from the supplier of ArduPilot, diydrones.com, so it takes more development time for implementing new sensors that are not offered. Many UAV research projects utilize ArduPilot (Bin et al, 2009) (Kabbabe et al, 2011) (Joseph et al, 2011).

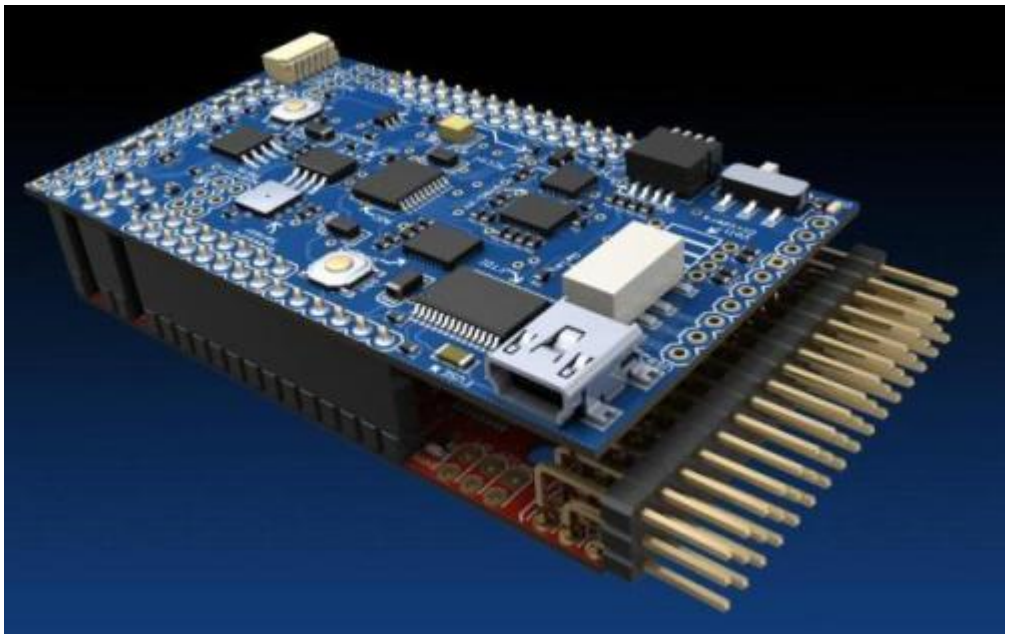


Figure 53: Ardupilot is an open source autopilot solution featuring arduinos.

3.1.1.3 Paparazzi

Paparazzi is an open source Linux project. The project includes software to run on open source hardware, which is available assembled from vendors. It will handle flying the plane and includes a ground base station for interfacing with the plane during flight. The paparazzi autopilot is designed to run on ARM7 and STM32 family of processors.

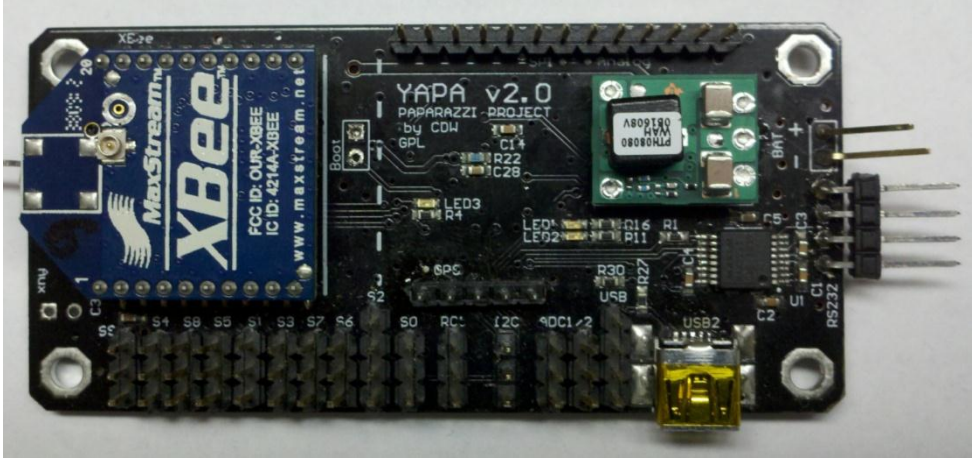


Figure 54: Paparazzi autopilot system. This particular version is named YAPA2.

Paparazzi allows for the configuration of many different sensor combinations for use with flying. New control loops can be adjusted or implemented based on user requirements. The advantage of the paparazzi project is that it allows the user to create custom subsystems to support sensors not within the list of sensor firmware available for the autopilot. Many autopilot research projects utilize the Paparazzi autopilot system (Reuder et al, 2009) (Bronz et al, January 2010) (Jensen et al, 2008).

3.1.2 Flight sensors

In order to determine the planes position, orientation, speed, and other flight characteristics sensors are used to calculate and relay back this information to the ground station. All of this information is required to provide the closed loop feedback for the control system directed by the autopilot software to keep the plane stabilized during flight. The following sensors were considered for use with the autopilot.

3.1.2.1 IR Thermopile

IR Thermopiles measure infrared radiation (IR) in the same spectrum as the IR reflected from the Earth. By arranging the thermopiles on the plane such that they face along the wings, the planes orientation relative to the earth can be calculated. When horizontal, all thermopiles will measure approximately the same amount of IR with exception of the vertical thermopile. As the plane turns and one wing faces the Earth, the thermopile will measure a greater amount of IR compared to the other thermopile facing the sky. The same principle can be applied for climbing and descending. The vertical thermopile completes the six degrees of freedom to determine the planes approximate orientation in 3D space.

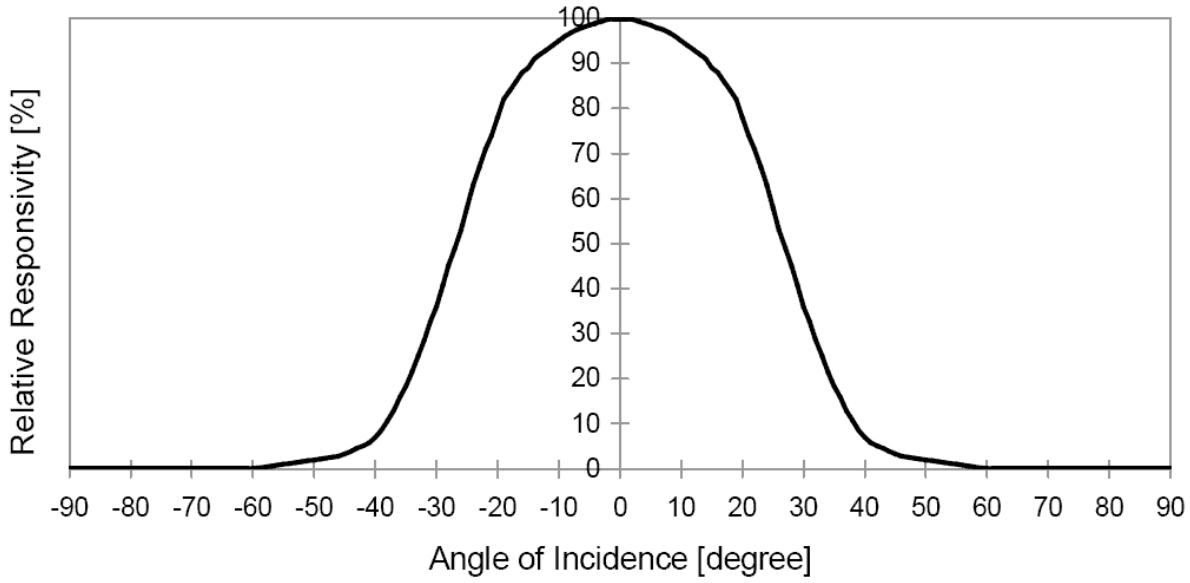


Figure 55: IR Response Curve. This curve represents the response of the thermopiles as the angle of the plane is changed. http://paparazzi.enac.fr/wiki/File:Ir_response_curve.gif

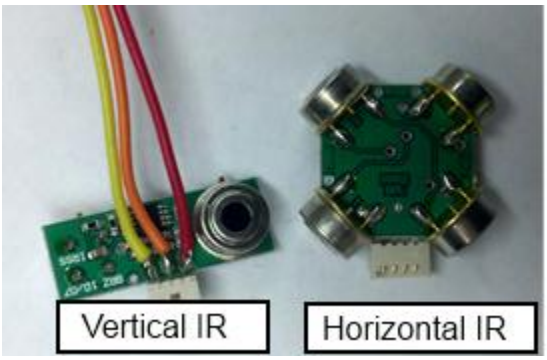


Figure 56: IR Thermopiles. They are in charge of providing flight stabilization for autopilot mode.

The thermopiles are arranged in pairs and wired through differential amplifiers to calculate the difference in IR readings. The voltage output from the amplifiers are wired to analog inputs located on the autopilot board.

3.1.2.2 GPS

The global positioning system (GPS) utilizes satellites orbiting around the Earth to obtain a position. A GPS receiver can be placed on the UAV to provide the position of the UAV during flight. Knowing the UAV's position and time, you can calculate the UAV's velocity. The flight plan will contain waypoints relative to the starting GPS coordinate. GPS modules are implemented on many different unmanned aerial research projects (Am et al,2007) (Abdelkrim et al,2008) (Chang-Sun et al, 2003).

3.1.2.3 Airspeed

By default the speed of the plane is calculated from the GPS. This works well in calm conditions but becomes inaccurate in windy conditions. Using an airspeed sensor will improve the planes throttle control to keep a constant speed, necessary for maintaining the location and altitude of the plane. Airspeed sensors are used on many different unmanned aerial research projects (Jung et al, 2005) (Beard et al, 2005) (Chao et al, 2010) (Beyeler et al, 2009).

3.1.2.4 Inertial Measurement Unit

The inertial measurement unit (IMU) is a combination of accelerometers and gyroscopes used to determine the acceleration rate of the airplane and changes in roll, yaw and pitch. Using time internally maintained by the processor, the velocity, position, and orientation of the aircraft can be calculated.

The IMU can be paired with some of the following sensors to compensate from possible accumulating errors. In addition to combining with additional sensors, the implementation of a khalman filter can be useful for determining the location of the aircraft over the duration of the flight. IMUs are used on many different unmanned aerial vehicle projects (Bayraktar et al, 2004) (Euston et al, 2008) (Dobrokhodov et al, 2005) (Kim et al, 2006).

3.1.2.5 Gyroscope

Gyroscopes measure changes in the UAV's rotation. This can be attached to the UAV to know its pitch, roll, or yaw to assist with the thermopiles stabilization. Although gyroscopes are contained within the IMU, additional gyroscopes can provide additional accuracy over the flight duration (Zhihai et al, 2005) (Metni et al, 2006).

3.2 UAV Control System Integration

Once the background was research, the next step was to determine which control system the team would use and the corresponding hardware to use. Once the hardware and control system were selected the control system was integrated with the platform system.

3.2.1 Control System Selection

To determine what hardware that will be needed to fly the plane the team needed to decide on which control system to use. Table 6 shows the decision matrix that was used to determine the control system. The decision matrix was weighted using the following criteria, which was assigned a value from 1-5. The values were assigned based on our background research and results from UAV competitions.

Value - how expensive (1 Most – 5 Least)
Availability - how available is the autopilot including hardware and software (1 Not – 5 Easily)
Versatility - how much of the autopilot is modifiable (1 Not – 5 All)
Implementation - how easy it is to implement the autopilot (1 Easy – 5 Difficult)
Ability - how powerful the autopilot is (1 Low Performance – 5 Great Performance)

Table 6: Decision matrix for determining which autopilot system the team used.

	Paparazzi	ArduPilot	Piccolo
Value	5	3	1
Availability	5	5	1
Versatility	4	2	1
Implementation	2	3	5
Ability	3	2	5
Totals	19	15	13

Based on the totals, the team decided on Paparazzi due to its versatility and immediate availability.

3.2.2 Hardware Selection

After the autopilot system was chosen, the microcontroller needed to be selected to run the autopilot software on the UAV. The microcontroller will contain the portion of the paparazzi software which controls the plane during flight.

3.2.2.1 Microcontroller

Paparazzi autopilot software can compile onto ARM7 LPC21xx series and STM32 series controllers. Within the Paparazzi documentation, several boards are recommended to run Paparazzi. The next step was to determine if the autopilot will run on a Paparazzi board, develop a board to fit the teams specification, or find a substitute board. Finding a substitute board would be the cheapest option for the team, however getting the proper hardware without extra capabilities from a third party would prove to be challenging. Developing our own board would be the most complete solution for our project as it can be designed to have exactly what would be needed. This option would be the best for the team in the long run, though would take the longest time to complete. Ordering a Papparazzi developed board would be the most expensive solution, though would guarantee that the autopilot will compile on the board and have at least the minimal requirements for flight.

Our decision matrix for autopilot boards was weighted based on the following criteria.

Value - how expensive (1 Expensive – 5 Least Expensive)
Versatility - how adaptable the controller is (1 Less adaptable – 5 More adaptable)
Availability - are the boards currently available (1 Unavailable – 5 Available)
Implementation - amount of work for implementation with paparazzi (1 More Work – 5 Ready)

Table 7: Decision between using a paparazzi autopilot board verses using a development board.

	Paparazzi Board	**Development Board**
Value	3	4
Versatility	3	4
Availability	5	5
Implementation	5	2
Total	16	15

Based on the results it was decided to order a Paparazzi developed board as it will be readily available and is known to work for our application, where a development board will require more research to determine the best fit. Once it was decided that the team would be working with a Paparazzi developed board it was narrowed down to three Paparazzi boards based on their current availability.

Table 4 shows the three optional boards and their specifications. Based on this chart, the team decided on the YAPA 2. It was chosen because it featured the largest number of servos and it was available at the time. The size and weight difference of the YAPA 2 compared to the other boards is insignificant compared to the design specifications of the UAV.

Table 8: Decision between which paparazzi autopilot board to use.

Paparazzi Controllers	Lisa /M	YAPA 2	TWOG
Microprocessor	STM32F103RE	ARM 7 LPC2148	ARM 7 LPC2148
#Servos	7	10	6
Analog Inputs	7	6	8
SPI	Yes	Yes	Yes
I2C	Yes	Yes	Yes
Weight	10g	23g	8g
Size	33mm x 56mm	80mm x 40mm	40.2mm x 30.5mm
Peripherals	Pressure Sensor	Xbee headers	
Price	$250	$125	$125

3.2.3 Autopilot Configuration

Flying a paparazzi autopilot system on an airframe requires all of the components of the autopilot system to be configured correctly. The autopilot software consists of five .xml configuration files which are compiled together and then loaded on to the autopilot controller. Figure 57 is a summary chart of the configuration files.

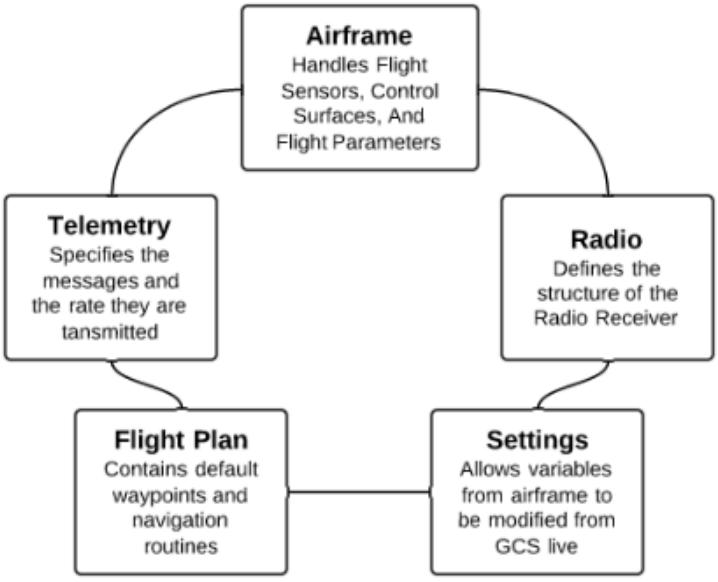

Airframe
Handles Flight Sensors, Control Surfaces, And Flight Parameters
Telemetry
Specifies the messages and the rate they are tansmitted
Radio
Defines the structure of the Radio Receiver
Flight Plan
Contains default waypoints and navigation routines
Settings
Allows variables from airframe to be modified from GCS live

Figure 57: The paparazzi configuration files allow for the complete configuration of the paparazzi autopilot system. This allows for many different types of airframe structures and sensor combinations to be implemented.

3.2.3.1 Airframe Configuration File

The airframe configuration file consists of the assignment of configuration files for all subsystems and sensors. Also included are the definitions of the control surfaces and their behaviors, and the defaults for many flight parameters. A configuration file system allows Paparazzi to be flexible with new sensors or other subsystems, by allowing the user to define new configuration files and associating them with their corresponding sensors or subsystems within the airframe file. All UAV's involved will need to set up the following subsystems and sensors in order to run the autopilot.

Xbee Radios

To confirm that the team has properly set up all of our sensor configurations XBee radios were used to transmit all sensor readings back to the ground station. See Appendix G User Manual: Xbee Connection.

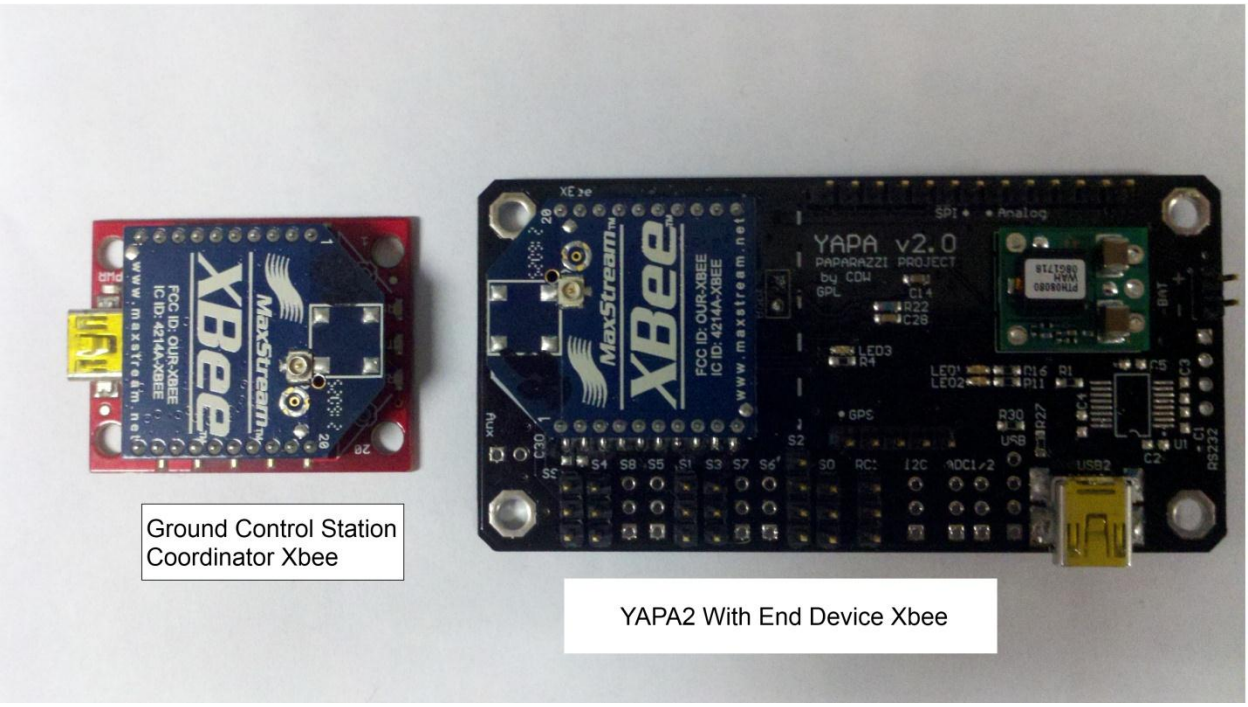


Figure 58 Paparazzi autopilot system showing the ground control station coordinator Xbee for connecting the ground station to the Xbee located on the Yapa2.

For multi-plane communications the Xbee radios had to be configured to use the Xbee-API-protocol which allows for local mesh networks. This involved updating the firmware for each of the Xbee radios used. See Appendix G User manual: Xbee Connection for more information.

Once the firmware was updated on the Xbee radios, the radios were installed on the two separate autopilot boards and communication was established to both simultaneously. See Appendix G User Manual: Multiplane Connection for more information. The aircraft identification number distinguishes the UAVs from one another on the ground control station end. The aircraft id number is defined in the configuration file for the list of optional aircrafts within the paparazzi main window.

GPS

With the ground station successfully communicating to the autopilot controller the GPS was interfaced with the autopilot controller next. To connect the GPS a 4-wire 100mil header that was soldered directly to the GPS adapter and then connected to the 5 pin header located on the board labeled GPS. Next the airframe configuration file had to be updated to communicate with the GPS.

```
<subsystem name="gps"          type="mediatek_diy"/>
```

Figure 59: Adding in GPS subsystem. This line of code from within the airframe configuration file will add the mediatek_diy GPS module to the airframe.

By defining the GPS subsystem, when the airframe is compiled it sees that the GPS subsystem is defined and it uses the settings defined by the type. The type in this case is the mediatek_diy which

corresponds to the GPS that was selected. The configuration file for a mediatek GPS was conveniently within the Paparazzi source code which saved time required to write a new firmware file to interface with the GPS.

Once the GPS was configured, the airframe file was compiled and downloaded to the autopilot controller. To confirm that the GPS was configured properly the ground station laptop, the autopilot controller, and the GPS were brought outside so that the GPS could get a signal from GPS satellites. After connecting to the autopilot controller from the ground station and then opening up the messages window, updated GPS coordinates from the autopilot controller were viewed in the window. These coordinates were in a universal transverse mercator (UTM) format and could be converted to latitude and longitude.

Figure 60: The mediatek GPS located at the front of Robin.

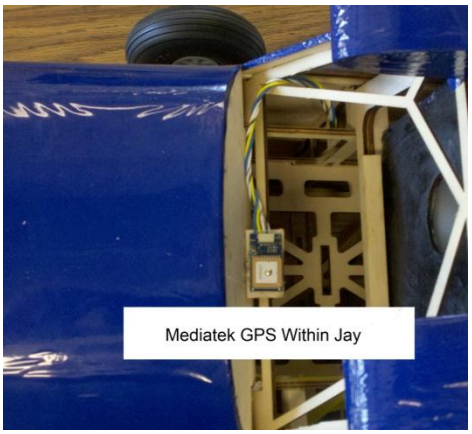


Figure 61: Mediatek GPS located at the front of Jay.

IR Thermopiles

Next the IR Thermopiles within the airframe file had to be configured. By loading the infrared_adc.xml file as a module, it arranged the ADC ports to sample the horizontal and vertical IR thermopiles. Loading the infrared_adc.xml file sets a make flag expecting the airframe configuration file to contain a section “INFRARED” shown in figure 62.

```
<section name="INFRARED" prefix="IR_">
    <!--Configurations for the infrared thermopiles-->

    <!--ADC neutrals-->
    <define name="ADC_IR1_NEUTRAL" value="512"/>
    <define name="ADC_IR2_NEUTRAL" value="517"/>
    <define name="ADC_TOP_NEUTRAL" value="514"/>

    <define name="LATERAL_CORRECTION" value="1."/>
    <define name="LONGITUDINAL_CORRECTION" value="1."/>
    <define name="VERTICAL_CORRECTION" value="1."/>

    <!--Sets if the horizontal is aligned with the wings or at a 45 degree angle-->
    <define name="HORIZ_SENSOR_ALIGNED" value="0"/>

    <!--Defines the signs for the different differential pairs-->
    <define name="IR1_SIGN" value="-1"/>            <!-- roll-->
    <define name="IR2_SIGN" value="-1"/>            <!-- pitch-->
    <define name="TOP_SIGN" value="-1"/>

    <!--Sets the roll and pitch neutrals, Roll is calibrated to prevent turning and the Pitch
        is calibrated to keep the UAV at a constant altitude while under throttle-->
    <define name="ROLL_NEUTRAL_DEFAULT" value="0" unit="deg"/>
    <define name="PITCH_NEUTRAL_DEFAULT" value="0" unit="deg"/>

    <define name="CORRECTION_UP" value="1."/>
    <define name="CORRECTION_DOWN" value="1."/>
    <define name="CORRECTION_LEFT" value="1."/>
    <define name="CORRECTION_RIGHT" value="1."/>
  </section>
```

Figure 62: The thermopile section within the airframe configuration file. This is where neutrals, signs and other parameters are configured.

Thermopile Calibration

Our first step to calibrating the thermopile was to find the neutral values of the IR differential pairs since thermopiles have varying neutral values due to the fabrication process. The IR sensors were placed within a foam box and clamped shut to isolate the sensors from any IR noise within the room. Within the foam box the IR sensors should register no IR readings so the value observed within the messages window of the ground station is an ADC bias reading. By adjusting the neutral values the ADC bias of the sensors can be removed.

Next the thermopiles were installed on the plane and the plane was brought out side to tune the roll and pitch neutrals. The UAV could not have any man made obstacles within 500-1000ft. First the neutrals were set to zero and placed the UAV on a level surface. Then the ground control station was turned began collecting readings from the autopilot. The neutral was set to offset the readings so that they are zero. Any readings not zero when the UAV is level needs to have a correct offset.

Next the UAV was tilted to a measured angle. The ground station reported the UAV's angle based on the IR readings. If the angle is incorrect the values "IR_360_Lateral_Correction" and "IR_360_Longitudinal_Correction" were defined and adjusted independently to obtain the correct angle.

The signs and corrections were used to change the sign of the reading of the thermopiles in case the sensor was mounted differently than intended. Once the mounting system was solidified these values did not have to change. The IR thermopiles were calibrated before each flight as they are the critical component for flight stabilization.

Control Surfaces

Our next step was to configure the control surfaces so that the autopilot can control the heading of the plane. The control surfaces fall under the commands section of the airframe configuration file. The commands section defines the name of the servos, which are connected to the control surfaces, how they are mixed together, and how they affect the roll, pitch, yaw, and throttle of the UAV.

```
<servos>
       <!--define the servo ports and their min/neutral/max values-->
       <servo name="MOTOR" no="0" min="1000" neutral="1440" max="1960"/>
       <servo name="AILEVON_LEFT" no="1" min="1000" neutral="1500" max="2000"/>
       <servo name="RUDDER" no="2" min="1040" neutral="1440" max="1840"/>
       <servo name="AILEVON_RIGHT" no="3" min="1000" neutral="1500" max="2000"/>
       <servo name="ELEVATOR" no="4" min="1880" neutral="1440" max="1040"/>
</servos>
```

Figure 63: Definition of the servos within the airframe configuration file. These definitions set the neutrals, minimum, and maximum positions for the control surfaces of the UAV.

The servos section defines the name of each of the servos, the channel number the servo is connected to, and the min, max, and neutral values the servos can be set too. The neutral value is the default value for the servo. See Appendix G User manual: Configuring Control Surfaces for more detail.

Flight Parameters

Before flying autonomously the flight parameters had to be defined. Flight parameters consist of two major sections titled “VERTICAL CONTROL” and “HORIZONTAL CONTROL”. The vertical control section contains parameters to set the limit for the throttle and pitch, as well as the default values for airspeed, ground speed, throttle, and pitch. The vertical control section also contains the values for the proportional and integral gains for the vertical control loop. Since these values are contained within the configuration file they will be easier to adjust. The horizontal control section contains limits for roll and pitch and their corresponding proportional and differential gains.

The settings configuration file allows for the modification of these variables during runtime from the ground control station which proved useful for tuning the control characteristics of the UAV during flight.

3.2.3.2 Telemetry

The telemetry configuration file defines the messages used for plane to base station communications. All of the messages within the telemetry are transmitted by the UAV and then displayed at the ground station within the messages window. This allows for debugging the control system and its various subsystems on the ground and during flight. All messages are saved for review

after the flight. Each type of message has a period defined. This allows for less important messages to be slowed down reducing data transmission.

```
<message name="AIRSPEED"            period="1"/>
<message name="ALIVE"               period="5"/>
<message name="GPS"                 period="0.25"/>
<message name="NAVIGATION"          period="1."/>
<message name="ATTITUDE"            period="0.5"/>
<message name="ESTIMATOR"           period="0.5"/>
<message name="ENERGY"              period="2.5"/>
<message name="WP_MOVED"            period="0.5"/>
<message name="CIRCLE"              period="1.05"/>
<message name="DESIRED"             period="1.05"/>
<message name="BAT"                 period="1.1"/>
```

Figure 64: Portion of the Telemetry configuration file. This shows some of the available types of messages that can be sent across the xbee radios along with the rate in which they are transmitted.

Figure 64 shows some of the messages that are sent from the default telemetry file. After testing the autopilot in the air unnecessary messages were removed from the default telemetry file. Some important messages that were used were the PPM and the command messages. The PPM message contains the last radio receiver packet and its values. The command message contains the target position of the servos. These two messages were key for determining how the radio encoder board operated. See radio control for more details.

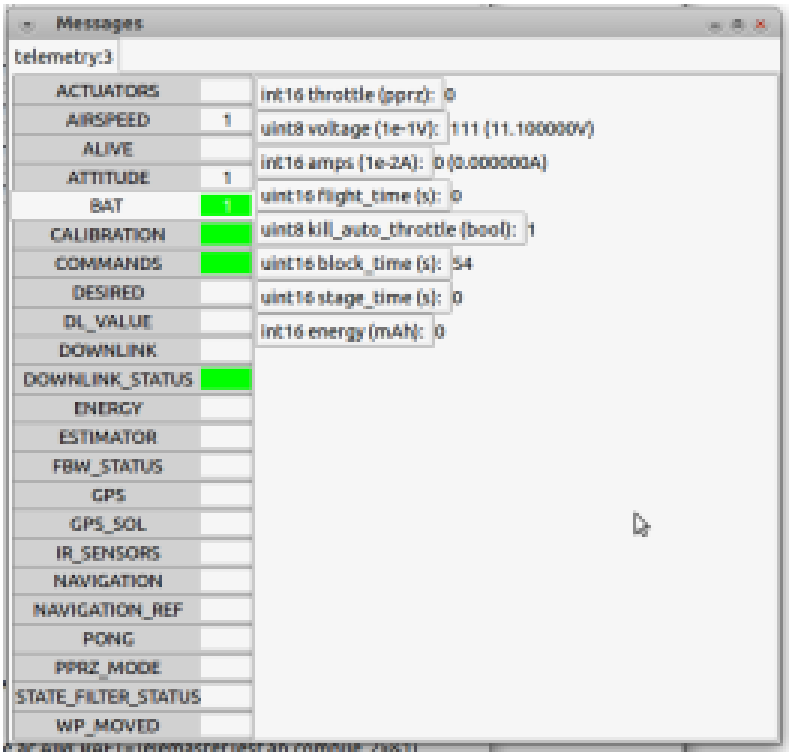


Figure 65: Above is an example of the messages window provided by the ground station to view incoming messages from the UAV. The "bat" message is currently selected and shows values such as voltage and current from the battery.

```
93.355 8 NAVIGATION 2 1 -2.4 4. 0. 21.9 0 0
93.375 8 GPS 3 26853456 468411392 2955 160370 37 -28 1662 252373750 19 1
93.423 8 WP_MOVED 15 268414.5 4684145.5 234.559998 19
93.424 8 CALIBRATION 0. 0
93.519 8 AIRSPEED 15. 14.5 14.5 6.
93.603 8 NAVIGATION 2 1 -2.4 4. 0. 21.9 0 0
93.631 8 GPS 3 26853456 468411392 2988 160290 34 -32 1662 252374000 19 1
93.632 8 NAVIGATION 2 1 -2.4 4. 0. 21.9 0 0
93.719 8 ATTITUDE 1.925012 5.238955 -1.318443
93.823 8 ESTIMATOR 0. 0.
93.831 8 DESIRED 0. 0.24 0. 0. 0. 45. 0.
93.855 8 NAVIGATION 2 1 -2.4 4. 0. 21.9 0 0
93.875 8 GPS 3 26853456 468411392 3001 160250 32 -16 1662 252374250 19 1
93.919 8 WP_MOVED 0 268579. 4684152. 234.559998 19
94.043 8 DOWNLINK_STATUS 94 32917 1471 0 469. 20. 17.28
94.103 8 FBW_STATUS 2 0 1 126 0
94.111 8 NAVIGATION 2 1 -3. 4. 0. 25. 0 0
94.131 8 GPS 3 26853400 468411392 3153 160160 31 -36 1662 252374500 19 1
```

Figure 66: Above is a sample data file collected during tested. The messages above were selected by the telemetry file and then recorded when received from the UAV.

3.2.3.4 Radio Control

Paparazzi autopilots provide a radio controller interface system for manual override of the UAV during flight. This is a critical safety feature for the autopilot and also required by FAA regulations for civilian UAVs.

The autopilot expects a pulse position modulation (PPM) from a radio controller's receiver. The radio configuration file defines each of the channels and the time ranges for each.

Some systems currently use a PPM communication system and it's as simple as wiring the receiver's PPM line to the R/C PPM-IN pin on the autopilot board. Unfortunately not all systems utilize PPM or the PPM trace is difficult to access on the receiver.

Our solution was the MeekpPE encoder board. It is an open source PPM encoder board, which takes the servo channels and mixes them into a PPM signal. This allows for a PPM signal to be generated from any style of receiver system and without modification of the receiver. See Appendix G User Manual: Radio Controller Interface for more details on PPM and how it's measured.

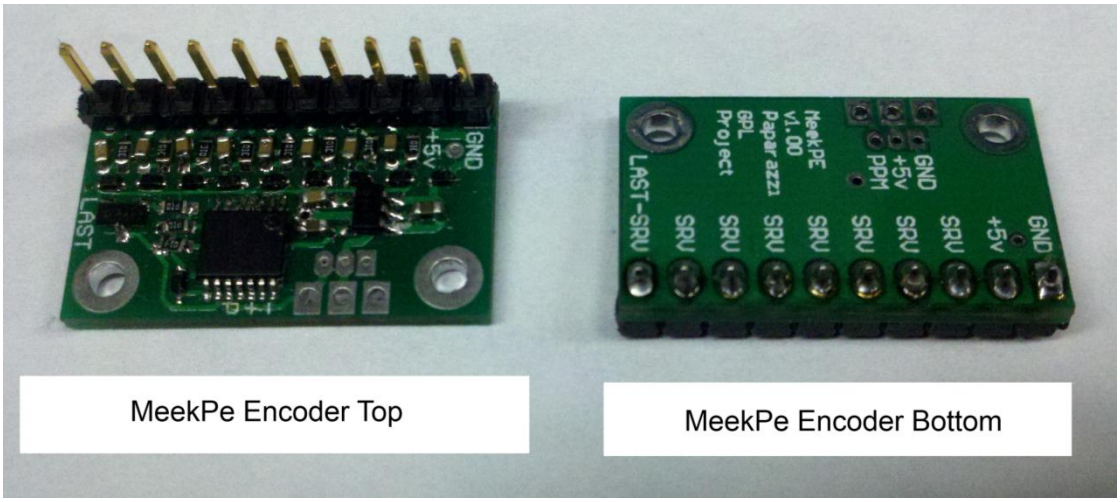


Figure 67: The top and the bottom of the MeekPe encoder board. This encoder board allows us to read the commands for the servo channels from the radio receiver without modifying it.

Acquiring this board put the autopilot testing behind by a few weeks. When the board was ordered it took the supplier over a month to arrive but this was due to some scheduled down time that the supplier. In the mean time while waiting for the board the team attempted to make our own variant of it so progress could be made. A few attempts were made to make our own and eventually had some success. One of our attempts involved making our own printed circuit board by following an etching process. This was attempted by downloading the gerber files and printing it on some photo paper using a laser jet printer. Once it was printed the image was transferred onto the copper using a hot clothes iron then etched away the rest of the material using an etching solution. The problems with doing this method were that the circuit used very small pitch components which couldn't be transferred with a high enough resolution image to the board. Unfortunately a mistake was made and the image was not mirrored before being ironed onto the copper. At that point we decide to scrap the printed circuit board and just made our own using standard through hole component from the ECE shop and a prototype board. This proved to be successful, and allowed the team to continue work with the autopilot system.

Following the steps described in Appendix G: User Manual: Radio Control Interface, it can be determined which servo channel is transmitted last. This last servo is wired to the Last-Srv via on the encoder board. Figure 68 is a block diagram demonstrating our wiring from the radio receiver to the MeekPe encoder board and then the block diagram of the MeekPe encoder board.

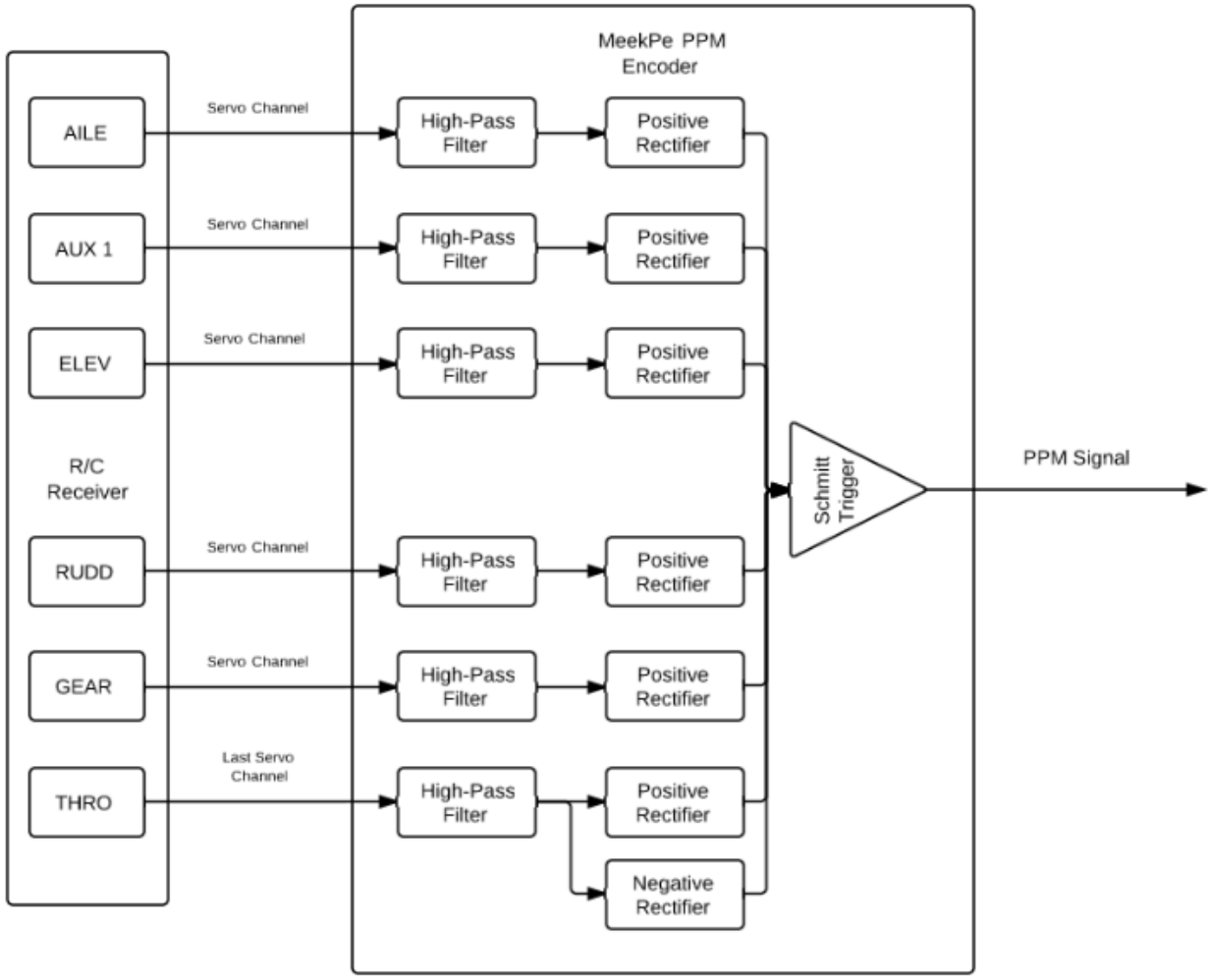


Figure 68: MeekPe PPM Encoder board block diagram. The encoder board takes in all of the servo commands and then sums them together into a PPM signal, which is easy to parse using the paparazzi radio configuration file.

Figure 69 is an image of the complete radio receiver system. Once assembled the radio receiver system was attached to the electronics mounting board. On the left side of the figure is the radio receiver and the right side is the MeekPe encoder board.

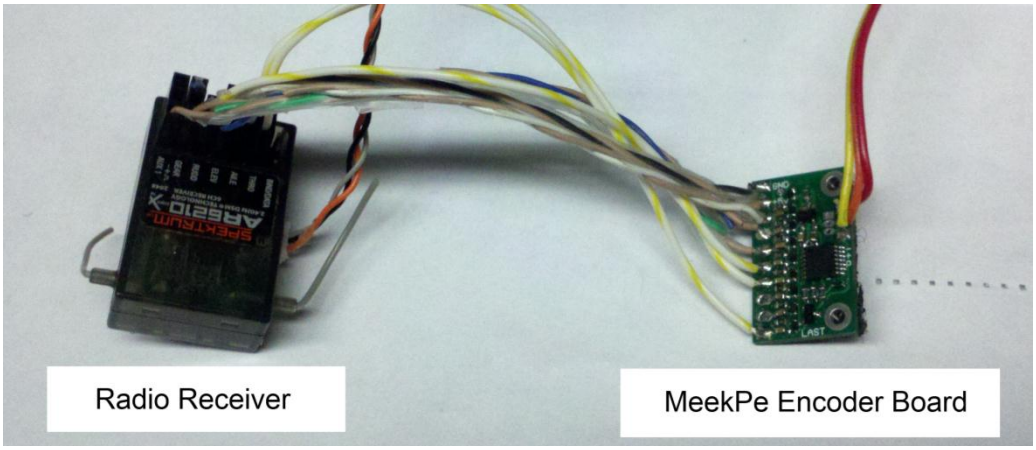


Figure 69: Radio Control System for controlling the plane through the autopilot system.

Radio Control Configuration File

The autopilot system uses the radio configuration file to parse the incoming PPM signal. Figure 70 is an example of a radio configuration file. See Appendix G User Manual: Radio Controller Interface for more details.

```
<!DOCTYPE radio SYSTEM "radio.dtd">
<radio name="AR6200" data_min="900" data_max="3100" sync_min ="5000" sync_max ="15000"
pulse_type="POSITIVE">
 <channel ctl="AIL" function="ROLL"       min="1220" neutral="1620" max="2020" average="0"/>
 <channel ctl="AUX" function="FLAP"       min="1220" neutral="1620" max="2020" average="0"/>
 <channel ctl="ELE" function="PITCH"      min="1220" neutral="1640" max="2040" average="0"/>
 <channel ctl="GEAR" function="MODE"      min="1100" neutral="1760" max="2060" average="0"/>
 <channel ctl="RUD" function="YAW"        min="2260" neutral="2660" max="3080" average="0"/>
 <channel ctl="THRO" function="THROTTLE"  min="1100" neutral="1520" max="2020" average="0"/>
</radio>
```

Figure 70: Radio Configuration File. This file defines the structure of the PPM signal received from the MeekPe encoder board.

The radio configuration file defines the channels seen within the PPM signal and parsed based on the parameters shown in Figure 70. The order of the channels is dependent on the receiver.

Autopilot modes

The autopilot has three main operating modes, which are selectable from the radio controller by the mode channel.

- Manual – Provides full control over all of the UAV's control surfaces as if it was a normal RC plane. This mode will be utilized for taking off and landing the UAV during initial autopilot testing. This mode is also FAA mandated.
- Auto1 - Engages the autopilot to fly the plane straight but still allows the user to turn the plane slightly and control the throttle and rudder. This will be our main method for testing the autopilot during its first few flights to ensure that the control system functioning properly.

- Auto2 - Fully engages the autopilot and has it follow the flight plan defined in the flight plan configuration file . This is the full autopilot for taking off, landing, and navigating way points. For the scope of this project the team intends to have the UAV navigate GPS waypoints only.

Using the radio controller it is possible to switch from Auto1 or Auto2 back into manual mode at any point. This system will ensure the safety of the flight crew and the UAV itself during autopilot testing.

3.2.3.5 Flight plan configuration file

The flight plan configuration file contains the steps the UAV will take during the Auto2 mode. First the way points are defined with a name and an x y coordinate. The x y coordinate is in meters relative to the home location, which is defined in latitude and longitude.

```
<waypoint name="HOME" x="0" y="0"/>
<waypoint name="STDBY" x="49.5" y="100.1"/>
<waypoint name="1" x="10.1" y="189.9"/>
<waypoint name="2" x="132.3" y="139.1"/>
<waypoint name="MOB" x="137.0" y="-11.6"/>
<waypoint name="S1" x="-119.2" y="69.6"/>
<waypoint name="S2" x="274.4" y="209.5"/>
<waypoint alt="30.0" name="AF" x="177.4" y="45.1"/>
<waypoint alt="0.0" name="TD" x="28.8" y="57.0"/>
<waypoint name="_BASELEG" x="168.8" y="-13.8"/>
<waypoint name="CLIMB" x="-114.5" y="162.3"/>
```

Figure 71: Waypoint Definitions within flight plan. These waypoints are example waypoints contained within the flight plan. The waypoints position is relative in meters from the home waypoint.

Figure 72: Visualization of the flight plan over Tanner Hiller, the local private airport the team tested the planes.

Next the sequential blocks are defined which contain the steps the autopilot will follow and will not proceed based on the conditional statements. For example the first step in a flight plan is "Wait GPS". The flight plan can also be used to add additional navigation routines.

```
    <block name="Wait GPS">
      <set value="1" var="kill_throttle"/>
      <while cond="!GpsFixValid()"/>
    </block>
```

Figure 73: Wait GPS block within flight plan. This is an example of programmable blocks or steps within the flight plan. This example will run at the beginning of AUTO2 and until the GPS fix is valid.

```
strip_button="Line (wp 1-2)" strip_icon="line.png">
      <call fun="nav_line_init()"/>
      <call fun="nav_line(WP_1, WP_2, nav_radius)"/>
    </block>
 <block group="extra_pattern" name="Survey S1-S2"
```

Figure 74: Line navigation routine callback button. This block within the flight plan configuration file creates a function callback to the line navigation routine.

The code within Figure 74 will add a button to the control strip within the ground control station in which it calls the navigation routine nav_line. The function takes in two waypoints and a circle radius. This will have the UAV fly in a straight line through waypoints 1 and 2 and then perform a U-turn with the given radius.

3.2.3.6 Settings

The settings configuration file allowed us to change values during flight time for numerous subsystems. This expedited the testing process as it allowed us to change flight parameters from the ground as opposed to landing the UAV, and uploading new changes between each flight. Values defined in the settings file are shown in the settings tab within the ground station.

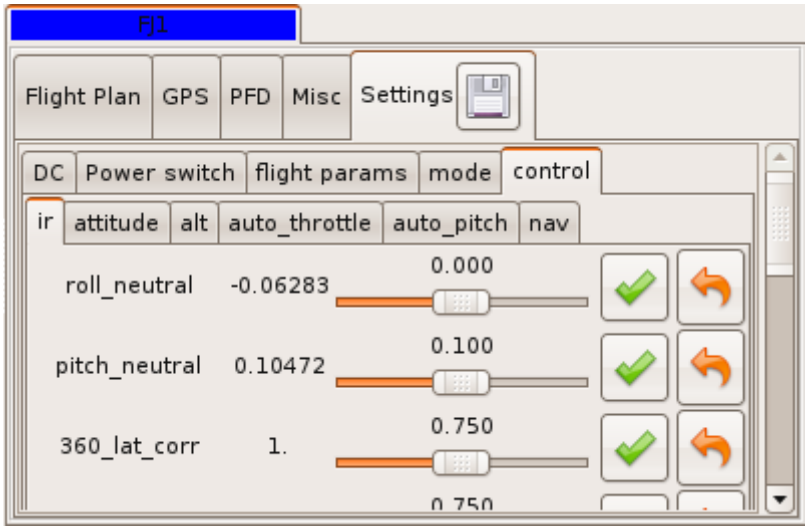


Figure 75: Settings Tab within Ground Station

3.2.4 Download and Operation

Once all the configuration files were completed, the autopilot software can be built and then downloaded to the YAPA2. This was an easy process using the paparazzi software. See Appendix G User Manual: Autopilot Connection for more details. Before ground testing began, simulations were ran on the configuration files to better understand how the system worked.

3.2.5 Autopilot Simulations

The paparazzi autopilot system provides the ability to run simulations on your airframe using the ground control station software. First the user builds a simulated version of the airframe configuration files, which defines the control characteristics of the plane, what the flight plan will for the simulation, what variables are changeable through the settings window, and which telemetry messages are sent. Running simulations on the airframe provided an initial feedback on how well the configuration files were configured, and showed ideal reactions after sending different commands to the autopilot. See Appendix G User Manual: Airframe Simulation for instructions on how to run simulations for more details.

3.2.4.1 Reading Simulation Data

Providing that the server program was running without the –n flag the data is stored within /paparazzi/var/logs titled with a date tag from when the simulation was started.

The messages stored in the data are defined in the telemetry configuration file. Each message has a time stamp and an aircraft id appended to the beginning. Figure 76 is an example of a GPS message.

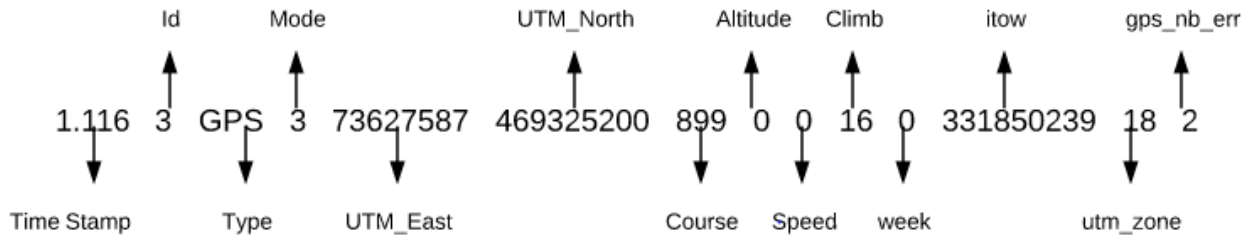


Figure 76: GPS Message Structure. The GPS messages from simulations contain the position of the aircraft, which was used to examine the effect of changing the parameters of the control loops.

The firmware for the GPS and the paparazzi system works with a GPS positioning protocol known as Universal Transverse Mercator (UTM). UTM involves separating the Earth into 60 zones where each has an origin. Worcester is located in zone 18, and the Tanner Hiller Airport is located at 73,627,587 meters east and 469,325,200 meters north of the origin of the zone.

3.2.4.2 Control Loop Simulations

Before beginning run simulations on the paparazzi system's control loops, a basic understanding of the controls behind UAVs was researched. The following controls equations are based on the article by Chang Boon Low in "A Trajectory Tracking Control Design for Fixed-wing Unmanned Aerial Vehicles". The state variables used in our control loops are [Heading, Velocity]^T these variables correspond to the servo position, and the output variables in our control loops.. A complete description of how each of these effect the motion of the airplane can be found in the Platform Background, "How to Fly" section. A general UAV kinematic equation of the motion can be modeled as:

$$\dot{x} = V * \cos\varphi$$
$$\dot{y} = V * \sin\varphi$$

Where x and y represents the position, φ represents heading, and V represents velocity. Variables with the subscript e are error terms. Variables with the subscript r are target variables. The tracking error that we want to stabilize in our Control equations is:

$$\begin{bmatrix} x_e \\ y_e \\ \varphi_e \end{bmatrix} = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \varphi_r - \varphi \end{bmatrix}$$

By differentiating this equation next the following tracking error model, given that ω is equal to the angular velocity was obtained:

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\varphi}_e \end{bmatrix} = \begin{bmatrix} \omega y_e - V + V_r \cos\varphi_e \\ -\omega x_e + V_r \sin\varphi_e \\ \omega_r - \omega \end{bmatrix}$$

With $V_L = \frac{1}{2}x_e^2 + \frac{1}{2}y_e^2$, the derivative along the following tracking error model can be solved to get:

$$\dot{V}_L = x_e(\omega y_e - V + V_r \cos\varphi_e) + y_e(-\omega x_e + V_r \sin\varphi_e)$$

Then by choosing $V = k_x x_e + V_r \cos\varphi_e$ and $\varphi_e = \tan^{-1}(-k_y y_e)$, where k_x>0 and k_y >0 , $\dot{V}$ becomes negative definite. By applying Lyapunov stability theorem we can conclude that the system is stable and (x_e, y_e) -> 0 as t -> 0, since V_L is positive definite while its derivative is negative definite. This is tracking control law is a continuous, static control law, and simply describes a simply control theorem to UAVs.

Using the GPS messages from the simulation, the team was able to simulate different responses from the control system after modifying PID constants for various control loops of the autopilot. The paparazzi autopilot control system can be summarized by the flow chart in figure 77.

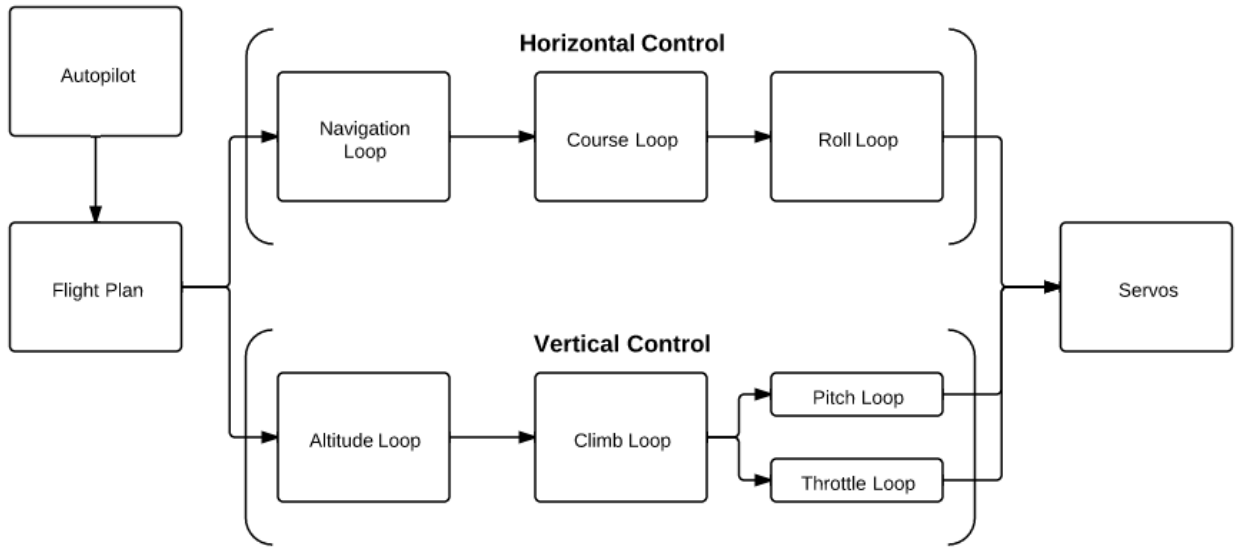


Figure 77: Paparazzi Autopilot Control Loop Structure. The overall structure features two main control loops, one for vertical, and one for horizontal, where the final output of each control loop define target positions for the control surfaces, driven by the servos.

The flight plan contains different navigation routines, which guide the autopilot through various target coordinates. Some navigation routines, such as standby, only contain one target waypoint and a set circling radius for that waypoint. A waypoint contains target positions, which set target positions for the horizontal control loop and the vertical control loop.

Vertical Control Loop

The vertical control is in charge of reaching target altitudes by controlling the target position of the elevator and the throttle of the engine as both have an effect on controlling the altitude of the plane. The vertical control loop is broken down by the flow chart in figure 78.

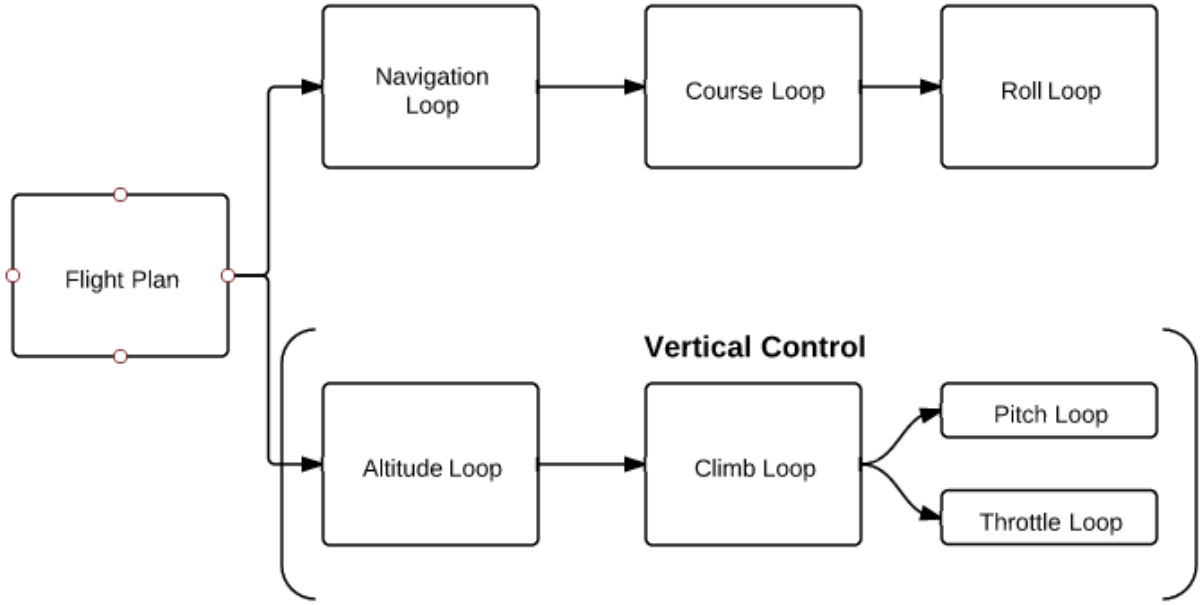


Figure 78: Vertical Control Loop

The vertical control loop's input is the target altitude. The target altitude is fed through two separate control loops, which control the throttle of the plane and the pitch of the plane. Both the throttle and the pitch can affect the change in altitude of the plane.

First, the team simulated the effect of changing the derivative constant of the pitch control loop to examine its effect on the takeoff of the plane during the simulation. The pitch control loop is depicted below in figure 79.

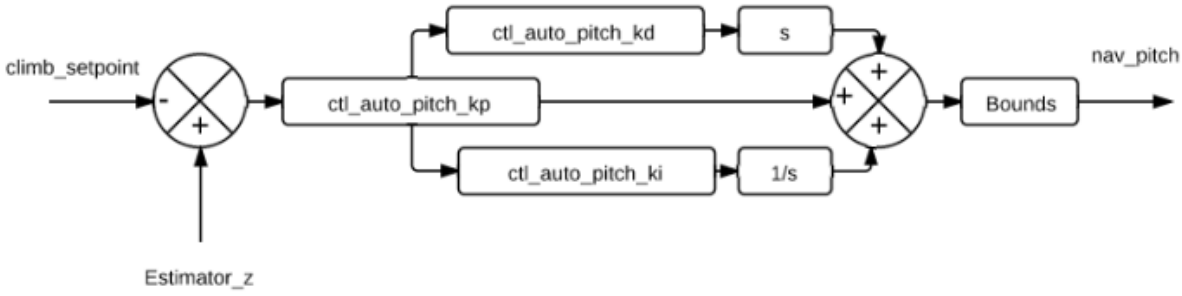


Figure 79: Pitch Control Loop

The output of the climb control loop is the climb_point, which defines the target climb point. The climb_setpoint is the input to both the throttle control loop and the pitch loop seen above. The structure of the pitch control loop contains proportional, integral, and derivative terms, which are summated into the resulting nav_pitch, which is limited by defined bounds. Nav_pitch is the desired pitch of the plane during flight. The plane's altitude is continuously calculated by the estimator, which

contains an estimation of the planes position and orientation based on available sensors. This acts as the feedback term for this system.

Nav_pitch is the input to the control loop overseeing the target set point for the elevator as the elevator is the main control surface for controlling the planes pitch. The elevators control loop is illustrated in Figure 80.

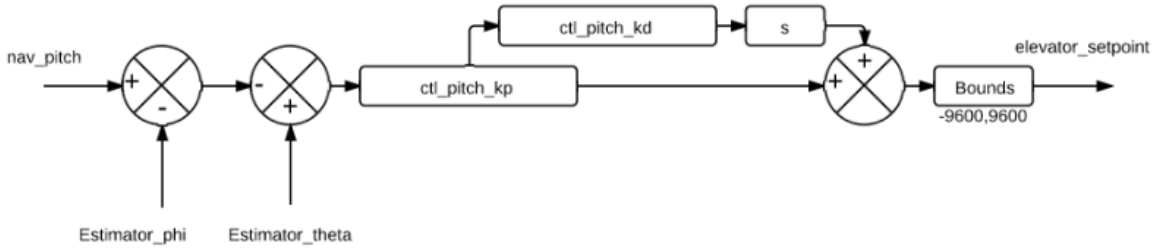


Figure 80: Elevator Position Control Loop

Shown in figure 80, the input to the elevator position control loop is nav_pitch, the output of the pitch control loop. The output is the target elevator position. The feedback of the system is the estimator's phi and theta value which represent the current orientation of the plane.

For our pitch simulations, the team modified clt_auto_pitch_kp and clt_auto_pitch_kd , which was part of the pitch control loop. The control constants are defined within the airframe configuration file and were modified before each simulation. The simulation started with the plane on the ground and then issued a command to take off. The first simulation was ran with the default settings for the configuration file with a kp of -20000, kd of 0 and a ki of -0.25. Figure 82 was captured from the end of one of the launch simulation.

Figure 81: Screen capture of the ground control station during a launch simulation

The launch simulation involved starting the plane at the home position. After launching the plane, the target position is set, and climbs in altitude until it hits 25 meters. Once the plane hits 25 meters, the plane enters the standby navigation routine where the plane sets the target altitude to 75 meters and then circles the standby coordinate with a radius of 80 meters.

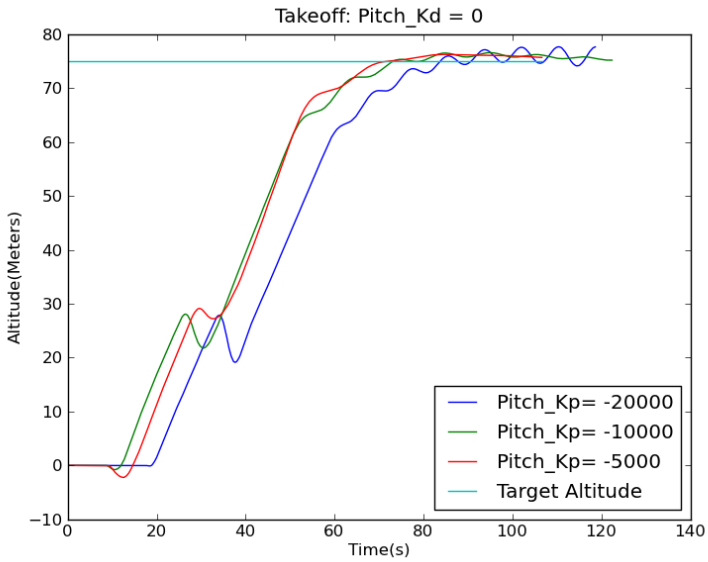


Figure 82: Pitch Launch Simulation where Kd = 0. This graph shows three different simulations where the kd constant was kept at 0 and then the kp constant was varied as represented in the graph key.

The altitude of the plane during the simulation was recorded within the GPS messages. A python script was used to parse the data for GPS and Altitude and then the data was plotted as shown above. After takeoff, the plane navigated to a waypoint with a target altitude of 75m, shown as the horizontal line at 75 meters. Without kd, increasing kp from -20000 to -5000 resulted in a more damped altitude response. For the next set of simulations kd was reduced to -500 and produced an expected response.

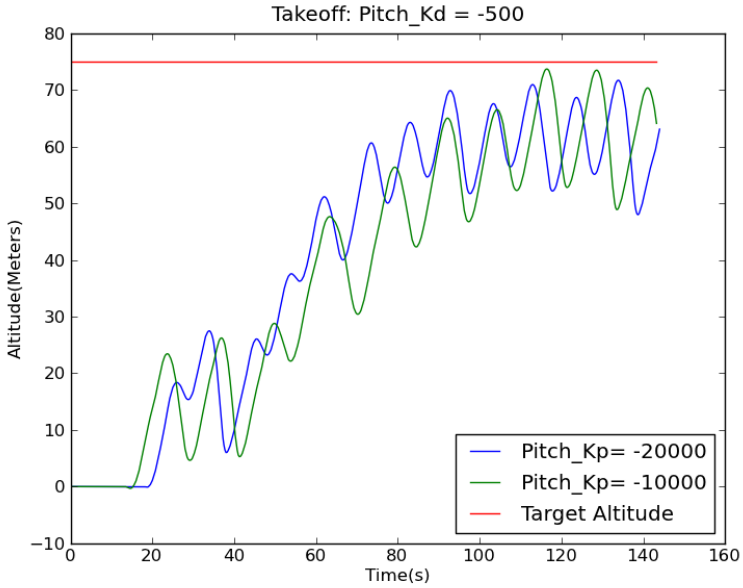


Figure 83: Pitch Launch Simulation Kd = -500. The graph above shows three simulations where the kd constant was kept at -500 and then the kp constant was varied.

In Figure 83, decreasing kd resulted in a more oscillatory launch. The proportional constant started at -20000 then reduced to -10000 as before but the third simulation with kp of -5000 failed to launch.

For the final pitch simulation, the derivative constant was increased to 500. The resulting graph is illustrated in Figure 84.

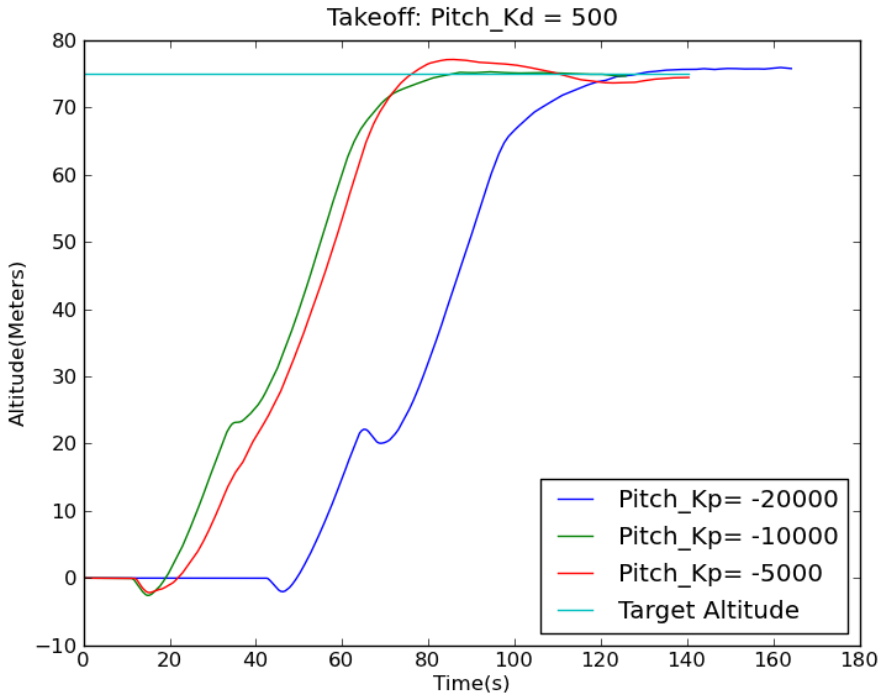


Figure 84: Pitch Launch Simulation Kd = 500. The graph above shows three simulations where the kd constant was kept at 500 and then the kp constant was varied.

Increasing kd resulted in the slowest launch but possessed the least amount of oscillations. As kp was increased from -20000 to -5000, the simulated planes assent increased and resulted in an overshoot at kp -5000. As a result of these simulations, the baseline values for these variables will be set to a pitch kp of -10000 and a pitch kd of 500.

Horizontal Control Loop

The horizontal control loop oversees waypoint navigation. The overall horizontal control loop can be broken down into the navigation loop, the course loop and the roll loop. Figure 85 shows a flow chart depicting the structure of the horizontal control loop.

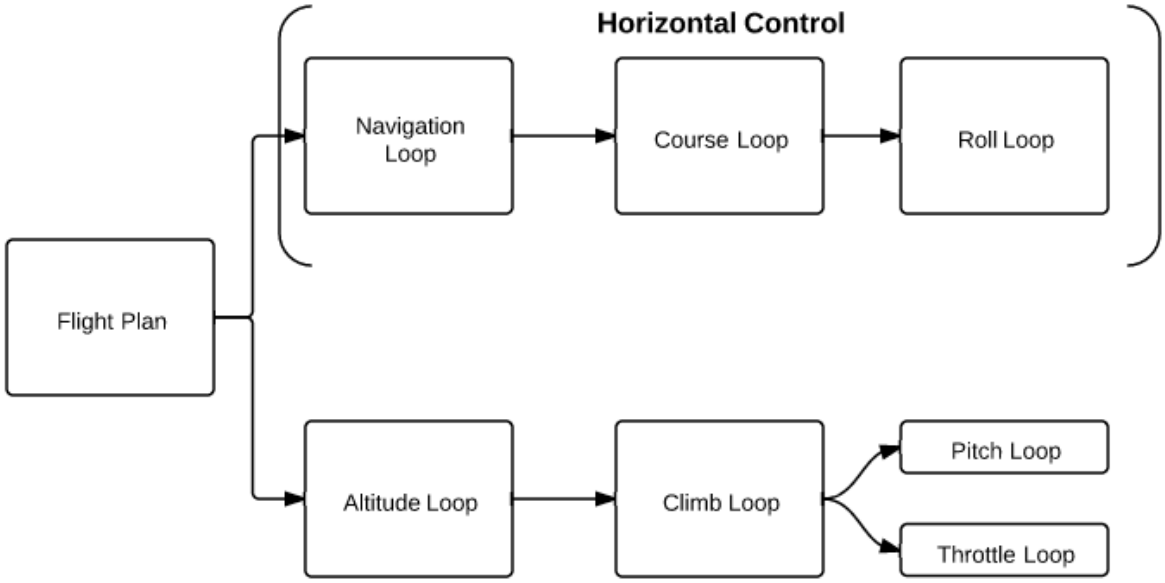


Figure 85: Horizontal Control Loop

The different navigation routines from the flight plan dictate the target waypoint for the plane. The course control loop's input is the course_setpoint which is determined from the current navigation routine.

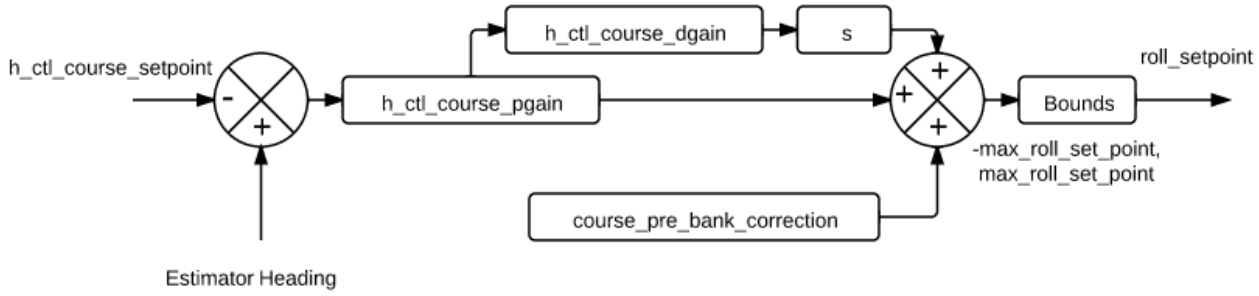


Figure 86: Course Control Loop

The course control loop shown in figure 86 generates a roll _setpoint based on the current heading and the target heading. A course_pre_bank_correction can be added to compensate for consistent errors in turning such as inconsistency in the servos or imperfections in the wings. Roll_setpoint is limited by a max roll set point. This prevents the plane from performing barrel rolls during flight. The roll_setpoint is the input to the roll control loop which is in charge of setting the planes bank angle.

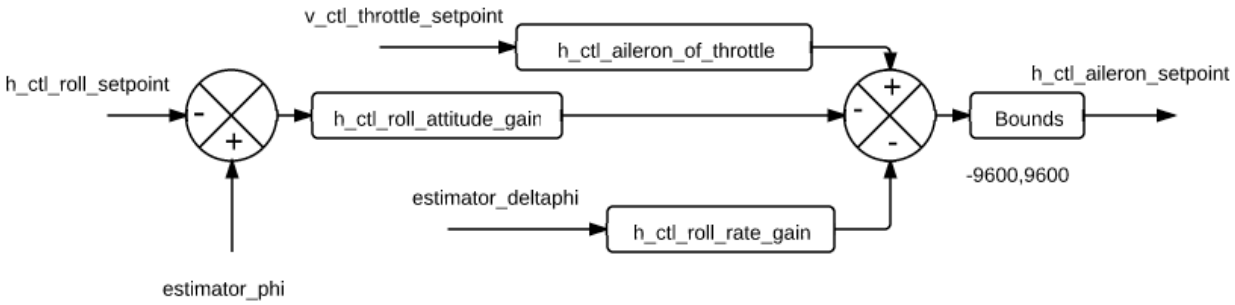


Figure 87: Roll Control Loop

The roll control loop sets the target position for the ailerons, the control surface with a direct effect on the roll of the plane. The estimator_phi value is the current phi, otherwise known as bank angle in this case, is the feedback for the system. A gain for the roll rate can be defined, which is contained in estimator_deltaphi, and a gain based on the current throttle set point so that a relationship between the throttle and the roll set point can be made. For our navigation simulation, the roll_attitude_gain was changed within the roll control loop to examine its effect on waypoint navigation.

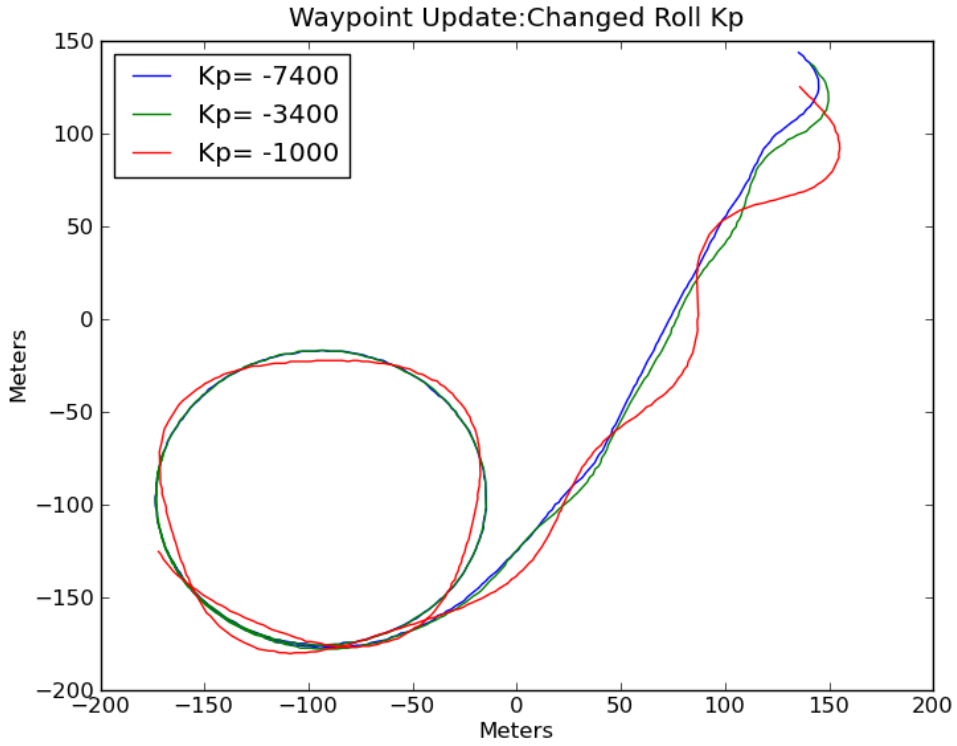


Figure 88: Waypoint Navigation Simulation

The simulated plane was launched and then circled the standby waypoint. As soon as the plane was on the far side of the standby waypoint, relative to the next waypoint, the command to circle the target waypoint was issued. The first simulation was started with a kp of -7400 which was the current

default value for the airframe. As the kp was increased from -7400 to -3400 minor oscillations can be seen when the waypoint was changed but the oscillations settled down as the plane approached the next waypoint. When kp was increased from -3400 to -1000 the system demonstrated an under damped response when changing waypoints and circling waypoints. The original and default kp generated the best response and will be used for autonomous flights. Figure 89 shows the end result of the waypoint navigation simulation.

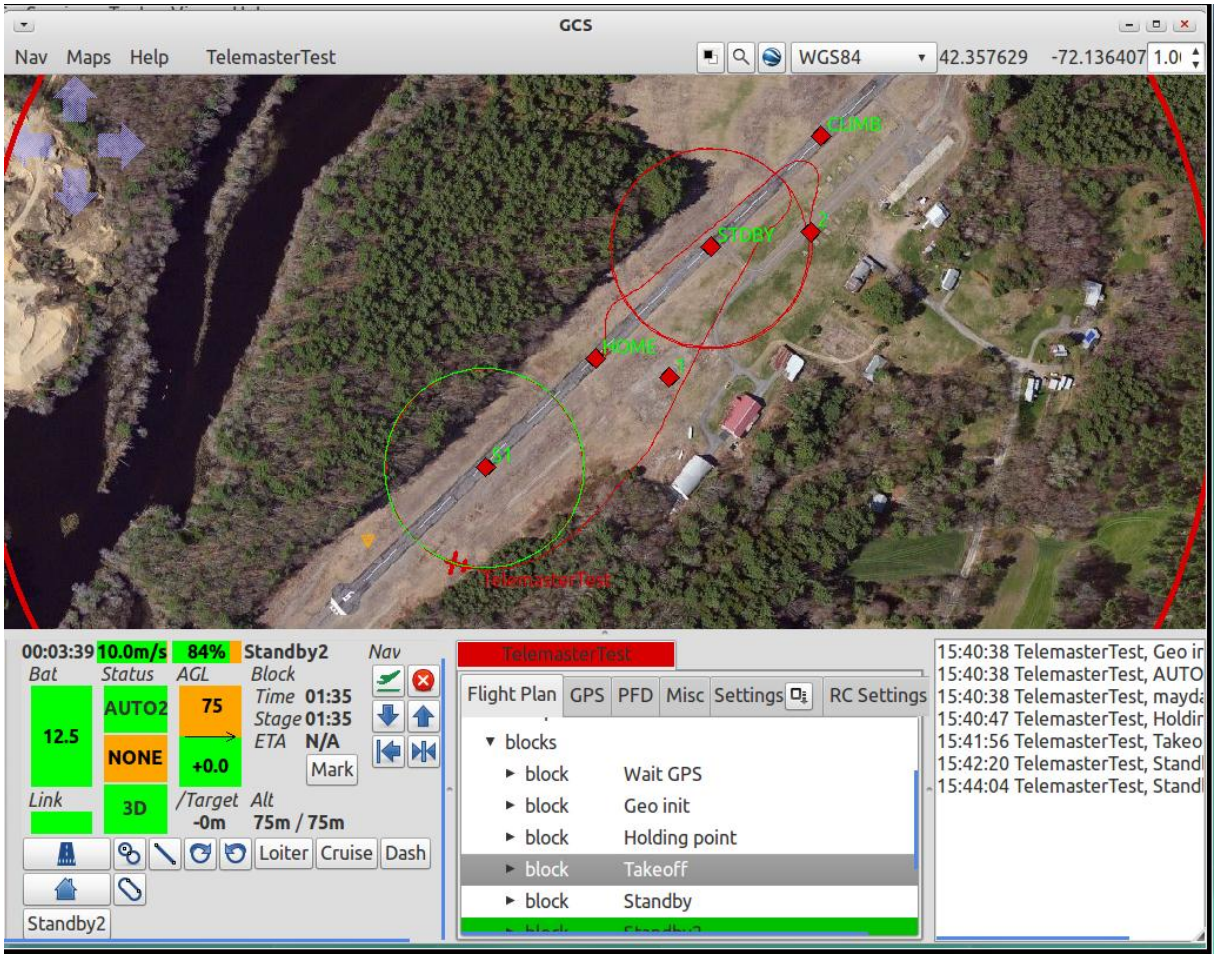


Figure 89: Waypoint Navigation Simulation. The goal of this simulation was to examine the effect of changing the roll_attitude_gain on waypoint navigation. When the plane was stable circling the standby waypoint the target waypoint was moved down the runway.

3.2.4.3 Adding Flight gear for visualization

An option that can be added to the paparazzi simulations is Flight Gear. Flight Gear is an open source remote control plane simulator. Adding Flight Gear to paparazzi provides a 3d visualization of the simulated plane flying. Figure 90 demonstrates a paparazzi simulation flying the plane within flight gear.

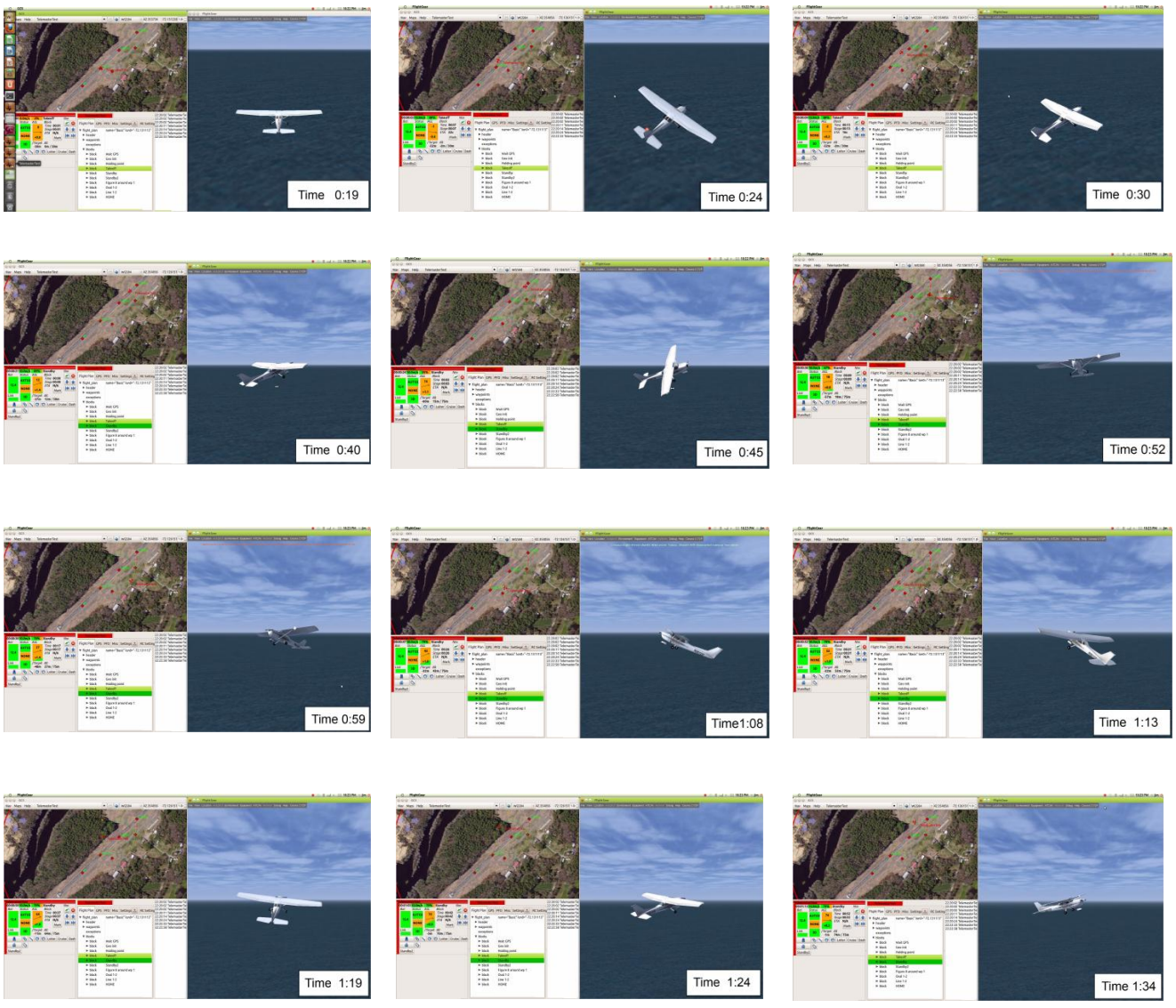


Figure 90: Flight Gear Simulation. This multi-image figure depicts the takeoff simulation of paparazzi alongside flight gear. Flight gear is an open source radio control flight simulator. The time of each picture was taken from the video is labeled.

3.2.4.4 Multi Plane Simulation

In addition to running single plane simulations, paparazzi has the ability to simulate and connect to more than one plane. Multiple planes can be simulated in the same fashion as a single plane. See Appendix G User Manual: Multi Plane connection for more details.

Once both of the airframes have been compiled, the user executes the simulation for one, and then selects the other airframe and then runs the simulation. This will generate a single ground control station for both airframes, where each can be controlled separately.

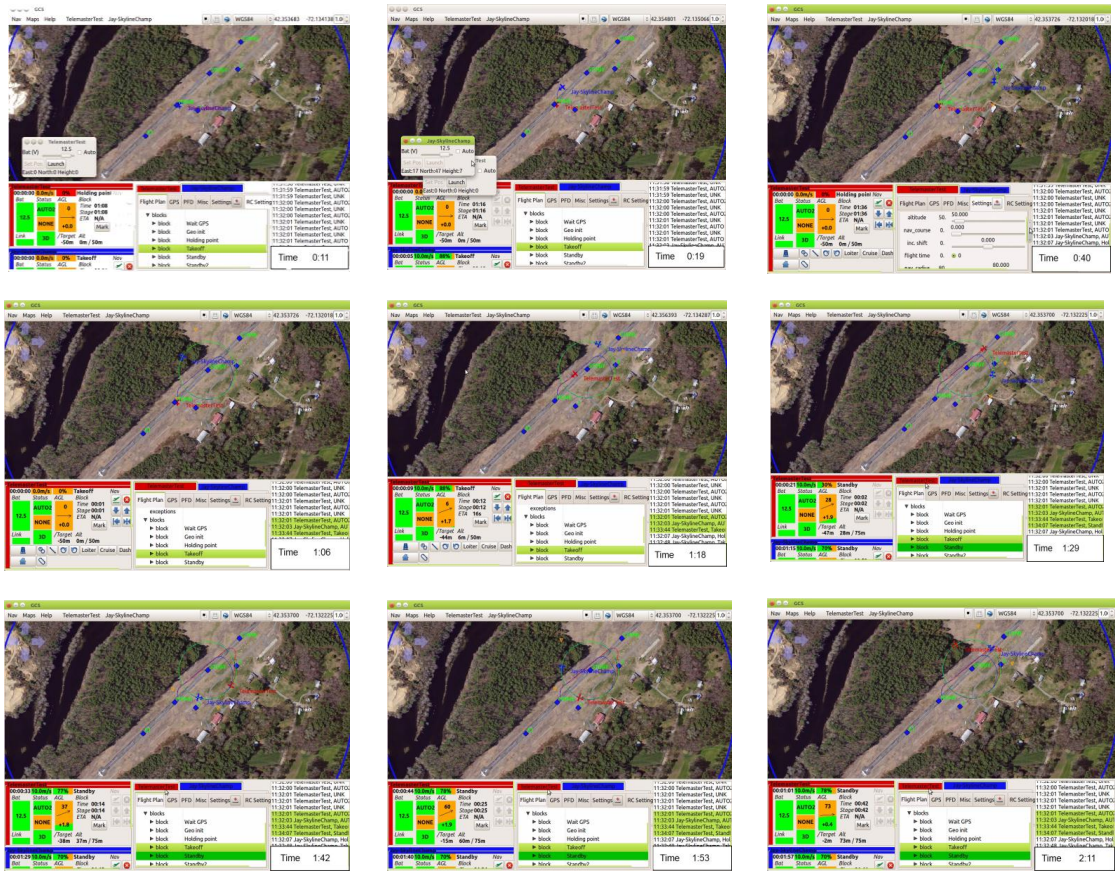


Figure 91: Multi-plane Simulation. This multi-image figure demonstrates a simulation in which two separate planes are flown. One plane is in blue and the other is in red. The time in which the snapshot was taken is located on the bottom right of each picture.

Running a simulation of two airframes allows for the visualization of a test plan featuring multiple airframes. For this simulation, each airframe was compiled with the same flight plan.

3.2.5 Autopilot Testing

After running simulations of the system, the team moved to physical testing of the autopilot system. Testing the autopilot started with testing on the ground within the lab, and then was moved

outside to examine the response of the IR thermopiles and the consistency of the GPS. Our plan for autopilot testing is explained in the following subsections.

Autopilot Ground Test

Once the airframe was confirmed operational via radio control test flights, we then installed the autopilot and its necessary sensors into the airplane. Before taking the UAV to the airfield for autopilot testing, we confirmed that all of the components of the autopilot were working as anticipated. This helped save debugging time on the field and prevented any unnecessary trips back to the field.

Within the lab we confirmed the operation of the following components:

- XBee communications:
    - Verified by connecting to the plane via the ground station software with the message window receiving messages
- GPS:
    - Verified by viewing messages received from the GPS within the messages window
- IR Thermopiles:
    - Verified by viewing the message window for IR Thermopile readings. The readings should be varied due to indoor IR noise.
- Manual Mode:
    - Verified by toggling the mode switch on the radio controller. Within the messages window and the ground station the mode should switch to Manual Mode. There should be full control over all of the control surfaces with the radio controller.
    - All of the servo throws have to match the servo throws produced by the radio receiver by itself. This is to ensure normal flight control with Manual mode
- Auto 1 Mode:
    - Verified by switching the mode switch to the Auto1 position and observing the plane. The control surfaces should attempt to stabilize the plane based on the IR Thermopile noise within the room. In addition the messages window and the ground control station should display that the UAV is in AUTO1.

Once the major components of the autopilot system were confirmed working within the lab we then proceeded to an AMA certified airfield or private airport for the autopilot assisted flight.

Autopilot assisted Flight

At the test field we will confirmed and tuned each of the major components of the autopilot before the flight. This procedure was the same within the lab with additional steps for the following components.

- GPS:
    - Should be able to obtain a GPS fix and the messages window should display correct coordinates from the GPS.
- IR Themopiles:

    - Before each flight, the IR Thermopiles must be recalibrated as the level of infrared radiation is never constant between different days. The thermopiles will be calibrated based on the process discussed in the infrared configuration section.
- Auto1:
    - The UAV should react as anticipated to being lifted and rolled based on IR Thermopile readings correctly as the thermopiles are now calibrated to the airfield.

For our first autopilot assisted flight we set the autopilot into Manual mode for a manual takeoff. Once there was a clear path for flying straight and safely away from the team we engaged Auto1. The plane was expected to fly in a straight line. Flight parameters were adjusted based on visual results.

The settings window was used to adjust values for the IR thermopiles and any of the neutrals for roll pitch and throttle to obtain a stable flight. In the event the UAV became unstable there was an operator on the radio controller ready to switch the UAV into manual mode and safely land the UAV.

Autopilot Navigating to Way Points

Once the team was thoroughly satisfied with the stabilization of the UAV during Auto 1, the team proceeded to use the full autopilot with Auto 2. When Auto 2 is enabled during flight the UAV will begin navigating the current command transmitted from the ground station. The default is to circle the standby waypoint defined within the flight plan. Circling will allow us to determine how stable the UAV is during turns as the radius of the circling can be adjusted. Once the UAV is circling the waypoint in an acceptable manner the final test will be the straight line navigation routine which would have the UAV navigate between two waypoints in a straight line with a U-turn after reaching the second point. Navigating between GPS coordinates was never achieved.

3.3 Autopilot Results

Using the open source paparazzi autopilot control system the team was able to succeed or fail in the following tasks:

Table 9 : Autopilot goals. This table lays out the autopilot goals and whether they were achieved, partially achieved, or not met.

Goals	Achieved	Partially Achieved	Not Achieved
Sensor Implementation	X		
Data Acquisition	X		
Multi-plane Connection	X		
Auto 1		X	
Auto 2			X
Waypoint Insertion From External Source		X	

3.3.1 Sensor Implementation

For use with the autopilot control software the GPS and the IR thermopiles were calibrated and integrated into the system. GPS data was used to track the planes position, and the IR thermopiles were in place for Auto 1 and 2 stabilization. An attempt was made to implement a compass module for data collection. GPS and thermopile readings are contained within the separate data files collected during testing.

3.3.2 Data acquisition

With the xBee radios streaming data back to the plane during RC flight tests, the autopilot software system successfully recorded data during several RC piloted flights. A sample of the telemetry data is displayed in Figure 92.

```
8.294 3 ATTITUDE -0.092519 1.090482 -0.133227
8.374 3 COMMANDS 8087,-4894,-2536,3079,-4894
8.430 3 NAVIGATION 2 1 11.1 -1. 0. 364.3 0 0
8.451 3 GPS 3 73641912 469324512 624 189260 14 -32 1674 417651750 18 1
8.474 3 FBW_STATUS 0 45 0 59 15076
8.594 3 AIRSPEED 5.291502 14.5 14.5 6.
8.682 3 PPM 45 24677,24677,24479,16605,39905,28805
8.686 3 NAVIGATION 2 1 11.1 -1. 0. 364.3 0 0
8.706 3 GPS 3 73641912 469324512 624 189180 11 -32 1674 417652000 18 1
8.711 3 NAVIGATION 2 1 11.1 -1. 0. 364.3 0 0
8.712 3 CALIBRATION 0. 0
8.778 3 RC -4894,-4894,-2538,-7545,3082,8084
8.793 3 ATTITUDE -0.120624 1.090482 -0.120624
8.874 3 COMMANDS 8080,-4895,-2534,3074,-4894
8.890 3 ESTIMATOR 189.193207 -0.174306
8.930 3 NAVIGATION 2 1 11.1 -1. 0. 364.3 0 0
8.950 3 GPS 3 73641912 469324512 624 189100 10 -32 1674 417652250 18 1
8.998 3 WP_MOVED 8 736437.8125 4693277.5 220.330002 18
8.999 3 IR_SENSORS -11 -14 -14 -11 99
9.004 3 DOWNLINK_STATUS 9 4420 208 1 522. 23. 14.42
9.099 3 DESIRED 0. 0.02 0. 0. 0. 50. 0.
9.182 3 PPM 46 24677,24678,24482,16606,39908,28816
9.186 3 NAVIGATION 2 1 11.1 -1. 0. 364.3 0 0
9.198 3 GPS 3 73641912 469324512 624 189030 8 -28 1674 417652500 18 1
9.278 3 RC -4896,-4895,-2532,-7547,3079,8083
9.378 3 COMMANDS 8087,-4891,-2536,3079,-4892
9.394 3 ESTIMATOR 189.070557 -0.192649
9.398 3 ENERGY 5.9 15.076 573 88.948402
9.430 3 NAVIGATION 2 1 11.1 -1. 0. 364.3 0 0
9.450 3 GPS 3 73641912 469324512 624 189070 17 16 1674 417652750 18 1
```

Figure 92: Data Acquisition during Manual Autopilot control. This portion of data shows RC messages with complete RC commands from the remote controller and updated GPS positions.

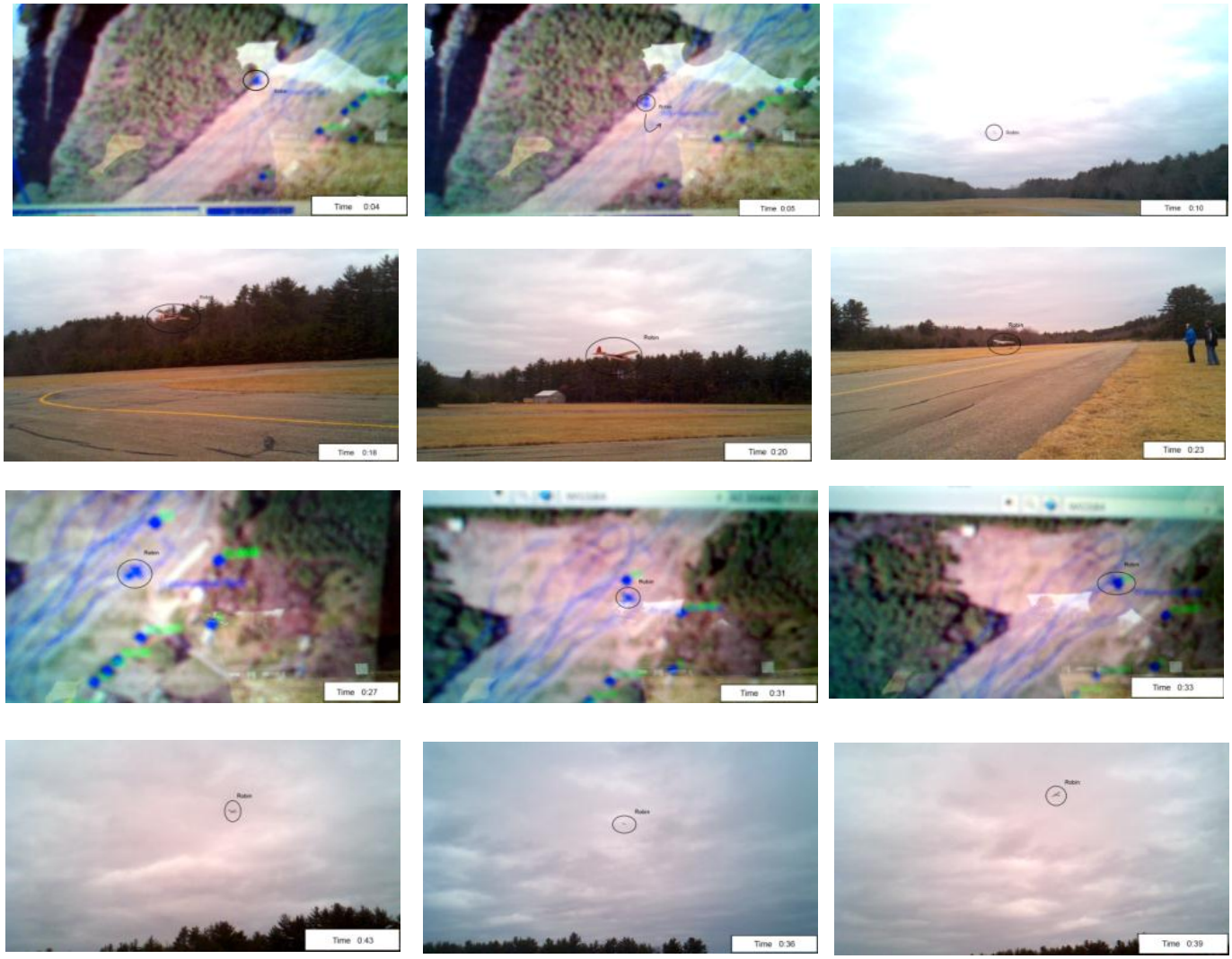


Figure 93: Autopilot data acquisition during RC flight. This figure alternates between the ground control station displaying the UAV's position and the UAV flying.

Figure 93 demonstrates a pure RC flight in which the autopilot was placed into the plane for telemetry transmission. The data was received from the autopilot within the plane was parsed to display the planes position within the ground station UI. The first two images display the plane coming around the end of the runway and then turning to come back down. The UI then shows the plane turning around again and then coming back.

3.3.3 Flight through the autopilot

Once the configuration file was complete and the radio encoder board was installed, the team successfully flown Robin on two separate occasions through the manual mode of the autopilot.

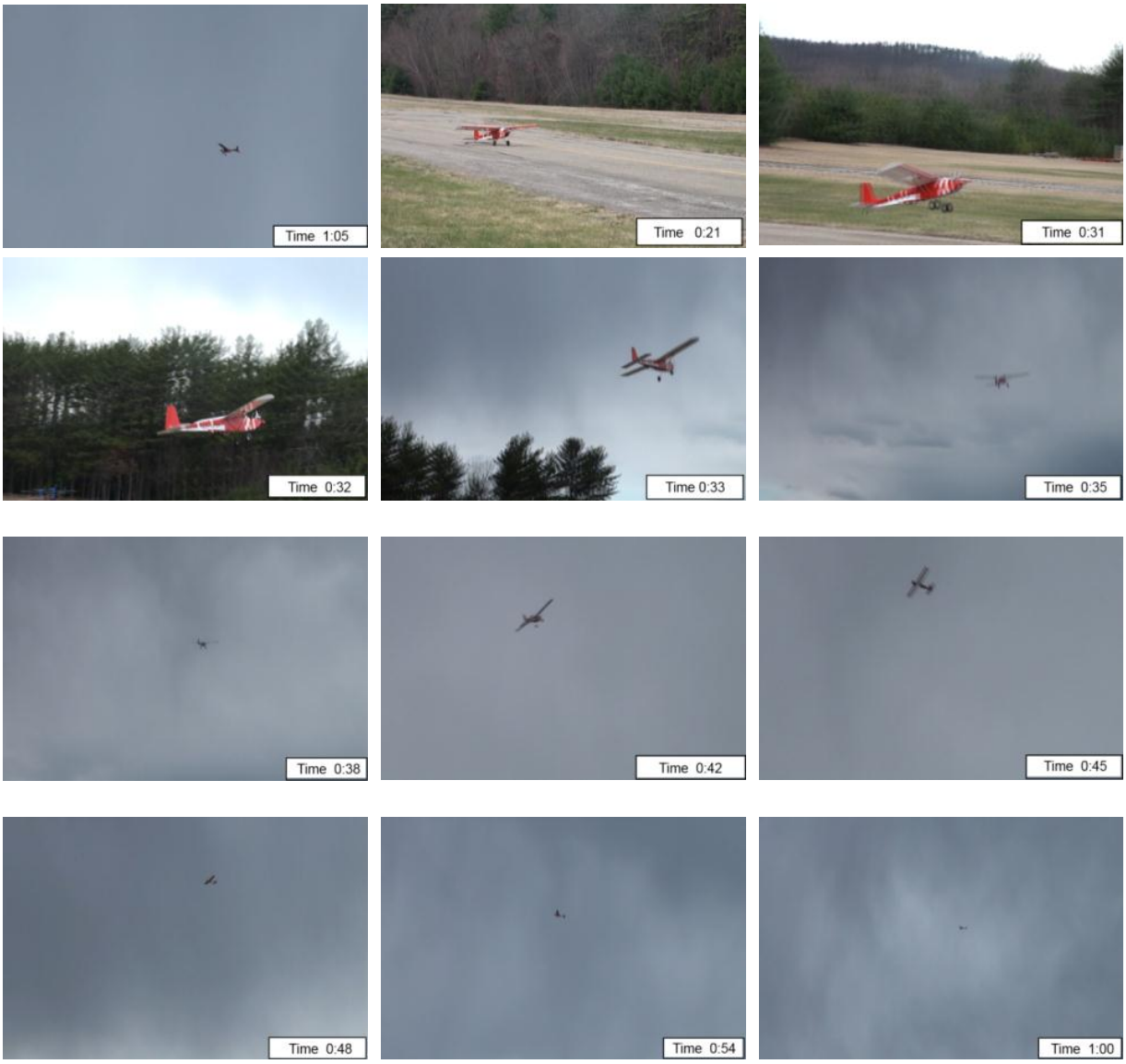


Figure 94: Successful takeoff using the manual mode of the autopilot.

Figure 94 demonstrates a successful take off of the plane using the manual mode of the autopilot. The above images are of the final takeoff of Robin before it's unfortunate crash.

3.3.4 Connection to multiple planes with data acquisition

After configuring the xbee radios into a Zigbee network, the team was able to connect to two separate autopilot controllers with separate aircraft id's to distinguish the two within the data and the ground control station.

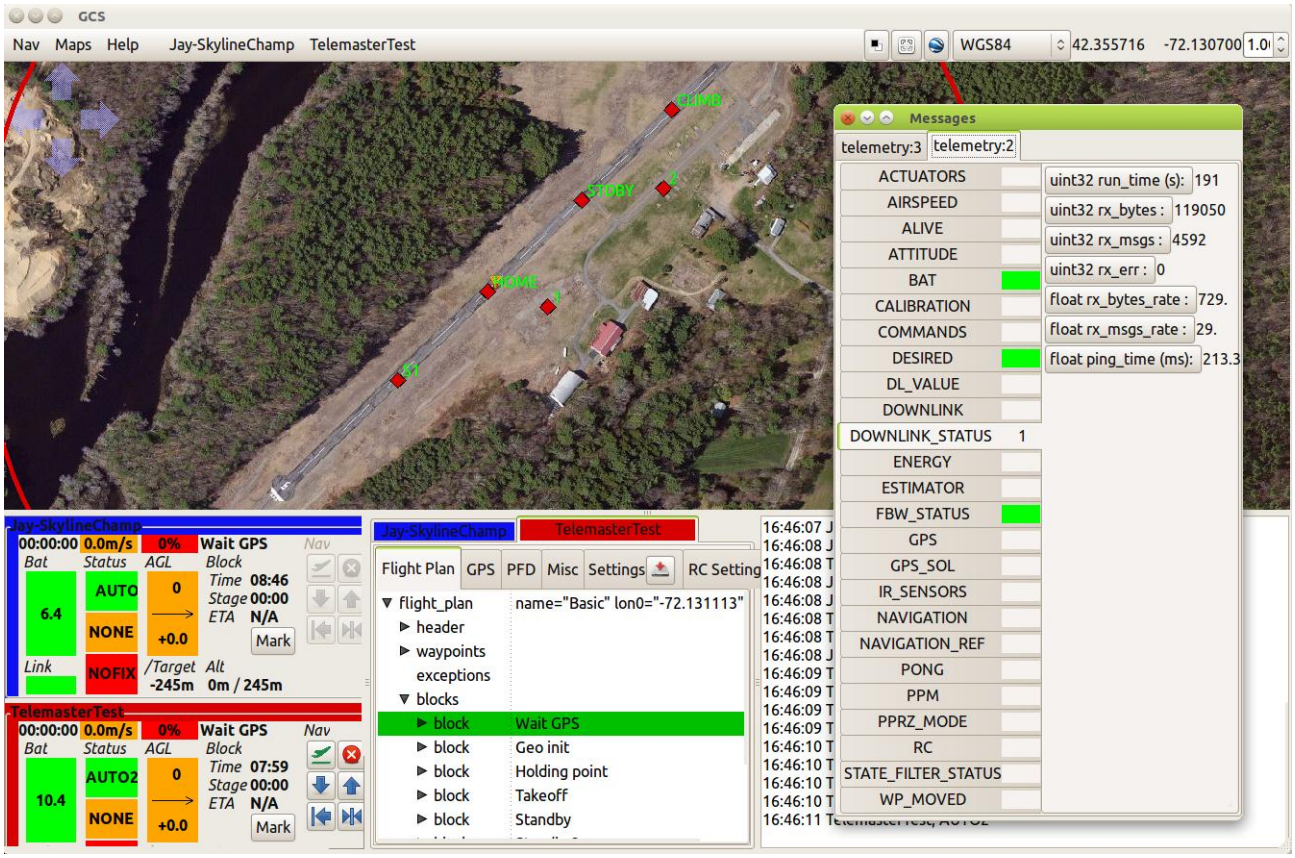


Figure 95: Multi-plane connection. This image demonstrates connectivity between two separate autopilot boards.

Figure 95 demonstrates the connection between two separate autopilot boards. The telemetry from both autopilot boards are recorded within one data file where the planes can be identified by its aircraft id number.

```
13.877 2 AIRSPEED 13. 12.5 12.5 10.
13.921 3 PPM 0 0,0,0,0,0,0
13.925 3 NAVIGATION 0 2 0. 0. 0. 86.1 0 0
13.931 2 PPM 0 0,0,0,0,0,0
13.941 3 GPS 0 0 0 0 0 0 0 0 0 0 1
13.962 2 GPS 0 0 0 0 0 0 0 0 0 0 1
13.964 2 NAVIGATION 0 2 0. 0. 0. 86.1 0 0
14.004 3 DOWNLINK_STATUS 28 16960 642 0 796. 31. -4981.75
14.005 2 DOWNLINK_STATUS 28 16328 625 0 681. 26. -4977.88
14.017 3 RC 0,0,0,0,0,0
14.033 2 RC 0,0,0,0,0,0
14.044 3 STATE_FILTER_STATUS 3 391
14.049 2 ATTITUDE 0.77941 0. 0.274627
14.117 3 COMMANDS 0,762,-9600,0
14.133 2 COMMANDS 0,779,-3334,0
14.136 3 ESTIMATOR 0. 0.
14.154 2 ESTIMATOR 0. 0.
14.168 3 NAVIGATION 0 2 0. 0. 0. 86.1 0 0
14.203 3 GPS 0 0 0 0 0 0 0 0 0 0 1
14.207 2 GPS 0 0 0 0 0 0 0 0 0 0 1
14.217 3 FBW_STATUS 2 0 1 104 0
14.239 3 WP_MOVED 3 736327.6875 4693233.5 235. 18
14.242 3 NAVIGATION_REF 736276 4693252 18
14.249 2 WP_MOVED 0 736318. 4693294. 235. 18
14.253 2 IR_SENSORS 249 71 71 249 253
14.257 2 CALIBRATION 0. 0
14.338 3 AIRSPEED 13. 12.5 12.5 10.
14.340 3 GPS_SOL 0 0 0 0
14.418 3 PPM 0 0,0,0,0,0,0
14.424 3 NAVIGATION 0 2 0. 0. 0. 86.1 0 0
14.440 2 PPM 0 0,0,0,0,0,0
14.450 3 GPS 0 0 0 0 0 0 0 0 0 0 1
14.453 2 NAVIGATION 0 2 0. 0. 0. 86.1 0 0
14.455 3 NAVIGATION 0 2 0. 0. 0. 86.1 0 0
14.465 2 GPS 0 0 0 0 0 0 0 0 0 0 1
14.467 2 BAT 0 64 0 0 1 364 0 0
```

Figure 96: Data Acquisition during Manual Autopilot control. This portion of data shows RC messages with complete RC commands from the remote controller and updated GPS positions.

Figure 96 shows data collected from two separate autopilot boards. The autopilot boards are distinguished based on their autopilot identification number defined within the ground control station. In this case the Jay autopilot configured board is identified as 2 and the Robin autopilot configuration is identified as 3. The airplane identification number is the second field within the telemetry data, directly next to the first field which is the time stamp relative to the start of the test in seconds.

3.3.5 Auto 1

After several successful RC flights on Robin through both RC control and Manual mode of the autopilot, the team engaged Auto 1 mode during a straight pass down the runway. When engaged, the plane banked to the right slightly. Due to an unfastened downlink antenna and lack of notice, we do not

possess telemetry data or footage of our Auto 1 test. Since we were only able to engage Auto1 once without calibrating the airframe and with data collection of the event the goal was considered partially met.

3.3.6 Auto 2

The overall goal for the system was to provide a system which flown between separate GPS waypoint s. This required a fully calibrated Auto1 for stabilized flight. Without Auto 1 testing we were unable to meet our goal of flying the plane in Auto 2.

4. Timeline

The original timeline created for the project is depicted in figure 97 in the form of a Gantt chart. This turned out to be a very optimistic timeline, and most of the milestone deadlines ended up being missed.

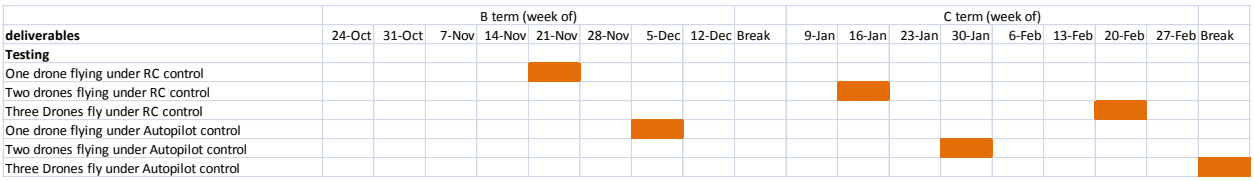


Figure 97: The General Gantt chart as predicted by the team in the beginning of October 2011

4.1 Timeline with regards to the Hardware team

The first major milestone in the testing section is the completion of the first airframe, ready for radio control testing, by the week of November 21. This milestone was completed on time, and there were flights in November of 2011 of Robin. The next testing milestone was scheduled for completion the week of December 5, 2011, and that was an autopilot test of the first airframe, Robin. This milestone was not met on time, the main reason for which was that there was no additional time given to properly tweak Robin to provide stable enough flight to give control over to the autopilot. The task of getting the airframe to consistently and predictably fly straight and level proved to be more of a challenge than originally anticipated. This required multiple trips to the test site with Mr. Gammond, the RC expert and pilot for the team. Each trip would result in multiple flights, and after each flight, Mr. Gammond would give a short briefing to the team regarding alterations that would improve flight performance, ultimately leading to the desired flight characteristics needed to ease the work the autopilot would need to do to keep the plane on course. For a complete list of flights and the results from each flight, refer to section 2.3. This list also illustrates how much work needed to be completed on Robin before an Auto 1 test was even considered. This extra work was the single most important factor that led to such a deviation from the original timeline. This fact also affected Jay and Duck, whose completion was scheduled for middle and end of C term, respectively. These deadlines were missed and thus far only Jay's airframe completion and RC flight milestones have been met.

Another setback regarding this was that the testing was scheduled to be done during the end of B term and into C term, which in Worcester, Massachusetts typically means difficult, cold winter

weather. The winter of 2011 – 2012 was abnormally mild, which made testing significantly easier, however, windy days often kept the airframes grounded.

4.2 Timeline with regards to Integration

A key aspect to this project was the integration between the different teams. The success of the integration is a defining factor of the success as the project as a whole. All of the teams acknowledged this fact early in the project, and as such the lines of communication were open from the start. Issues regarding power draw, weight and overall size of all of the components were carefully discussed between the teams to ensure the full system could be realized without any unrealistic requirements. For example, with power limited to that which could be carried by battery, it would be unrealistic to power a full Intel core I7 computer plus all of the other flight systems for any extended period of time. This allowed the teams to develop each portion of the project separately, while still allowing for integration questions to arise and be answered accordingly. This also allowed each project to move at its own pace without fear of interfering with the other projects. As such, there was never a set timeframe to perform the integration within the scope of the individual projects. The plan was to have each team finish their projects separately in C term, keeping integration in mind as the projects progressed through the year, and then after the projects were submitted the teams would convene and actually put all of the parts together in D term and do full system testing. This, due to the same setbacks as stated for the hardware team in particular, held the integration up as well. To see the actual steps towards integration, see section 6.

5. Budget

Project WiND was sponsored primarily by WPI and The Mathworks. The Mathworks donated $3,000 to all of project WiND, some of which was used on this project specifically. The final budget for all of the components used in this project is $2,770.94, the breakdown for which is located in Appendix E. This total is just a reflection of the purchases actually made. These purchases were decided based on necessity at the time of the purchase. For example, line item number 21 in the budget in the Appendix E is a DX6i radio system, of which only 1 is listed as being purchased. In order to implement the entire system, each UAV would be mandated by the FAA to have a manual override utilizing a radio system like this. Therefore, in order to have three UAVs flying simultaneously, there needs to be three completely separate radio systems present, one for each airframe. Since only one was purchased to save money, knowing that even if two UAVs were flight ready, a full system test with multiple UAVs flying simultaneously was not a short term possibility, and therefore the same radio system could be used for multiple UAVs at different times. There is a line item in the budget, line 6, called “9 Channel remote” for $89. This was purchased as a cheaper alternative to the DX6i early in the project, but this ultimately failed. Channel 1 was not transmitting properly, and it was outside the warranty. A replacement receiver module was ordered, but did not fix the issue. It was at this point that the DX6i was purchased.

Similar to the transmitter system, there are things that never got purchased that will be necessary for future testing, including a large capacity battery for powering the electronics for both

flight stabilization, control and servos, as well as the communications hardware, image processing hardware and cameras. Since this battery was never purchased, it is not listed in the final budget. The reasoning behind not purchasing the battery was its high cost as well as there was never a test scheduled that required that much power. In fact, there was only one flight that incorporated additional electrical components over the standard flight electronics, and the extra board did not draw enough power to warrant purchasing the larger battery at that time. We also only have two complete autopilot systems, which includes the autopilot board, the GPS sensor, the IR thermopiles, and the encoder board. We only have two complete sets because in January it was decided to focus on getting two of the UAVs airborne properly, rather than spread the existing funds too thin and not being able to successfully construct all three airframes. This also explains why there are only two gas tanks and only enough fuel line for two airframes. An additional gas tank and more fuel line will be required to finish the last UAV, Duck. For a complete list of what is left to purchase and complete on Duck, see section 2.2.6.

6. Integration with Software and Communications

Project wind is broken down into 5 distinct teams, three of which were located at WPI and the other two were at UNH. The hardware platform team who is the topic of this report features integration primarily with the software integration team.

For Project Wind, our teams goal was to provide a platform which can navigate waypoints that can be continuously updated using an easy to use command system. Software team would then generate the target waypoints to update the autopilot. For the software team we had to provide an USB for connection to the autopilot. It was software teams responsibility to ensure that the communications team could provide the final data structure needed. For updating waypoints the software team had to use the paparazzi protocol which consists of packed strings and floats with a checksum for error detection.

In addition to providing a command based system to update waypoints the platform team was responsible for housing all of the other team's electronics, meet their power requirements, and provide a camera gimbal system to house a camera for imaging. The hardware platform team made integration accomplishments discussed below.

6.1 External Waypoint Insertion

With a complete autopilot system, the goal was to integrate the autopilot control system with the Software team. This entailed providing the ability to externally control the autopilot in an easy manner. The goal was to have the plane follow the standby waypoint which is to be continuously updated by the software team's AI navigation path planning. Currently the autopilot system can be updated externally by utilizing the ivy bus protocol, which is currently used for communications between the autopilot board and the ground station.

Currently the autopilot board is connect to the IVY bus via a paparazzi source program titled "link.ml" . If this program is initiated, it is simple as shown in Figure 98 to update waypoints using the existing Paparazzi protocol. Unfortunately the paparazzi source language, OCAML, is currently not

supported by the linux distribution used by the software team on the mother ship. A new program will have to be designed to have the software team's boards connect to autopilot boards and then send commands to update waypoints.

```
p = pyproj.Proj(proj='utm', zone=18, ellps='WGS84')
lat,lon = (42.355648,-72.131113)
homeX,homeY = p(lon,lat)

lon,lat = p(homeX+x,homeY+y,inverse=True)

server = ivy.IvyServer("Python Server")
server.start()
time.sleep(0.1)

server.send_msg("ground MOVE_WAYPOINT 3 2 " + str(lat) +" "+ str(lon) + " 50")
```

Figure 98: Portion of the waypoint insertion script. This portion of code written in python inserted a target waypoint in x,y meters relative to home into a paparazzi instance.

As a proof of concept the software team generated a list of waypoints from their spiral navigation routine shown in Figure 99 and was inserted one by one into the paparazzi simulation using the python script mentioned in Figure 98.

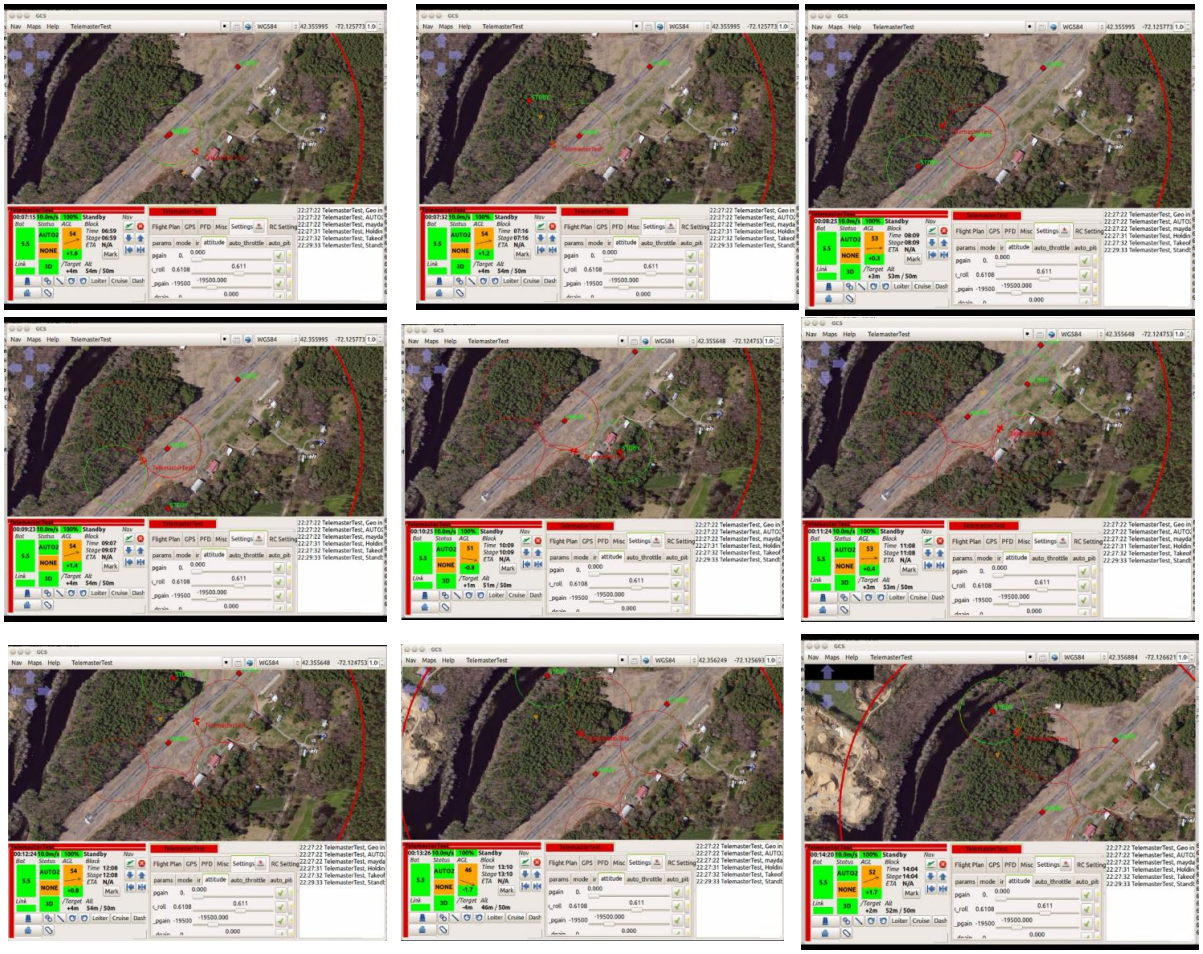


Figure 99: Waypoint Insertion Simulation. This sequence of images demonstrates the insertion of the software team's navigation routine into a paparazzi simulation.

Figure 99 shows the update of the target waypoint, in this case the standby waypoint, based on the output waypoints from the software integration team's navigation routine. Each image is taken upon assigning the next waypoint where the time of the simulation can be viewed in the panel on the bottom left corner of each image. From viewing the images left to right, path represents a spiral of circles around the center waypoint, This circling routine provides camera coverage over the target waypoint cell. Upon completion of the final design, the software integration component would have direct access to updating the waypoints from their system.

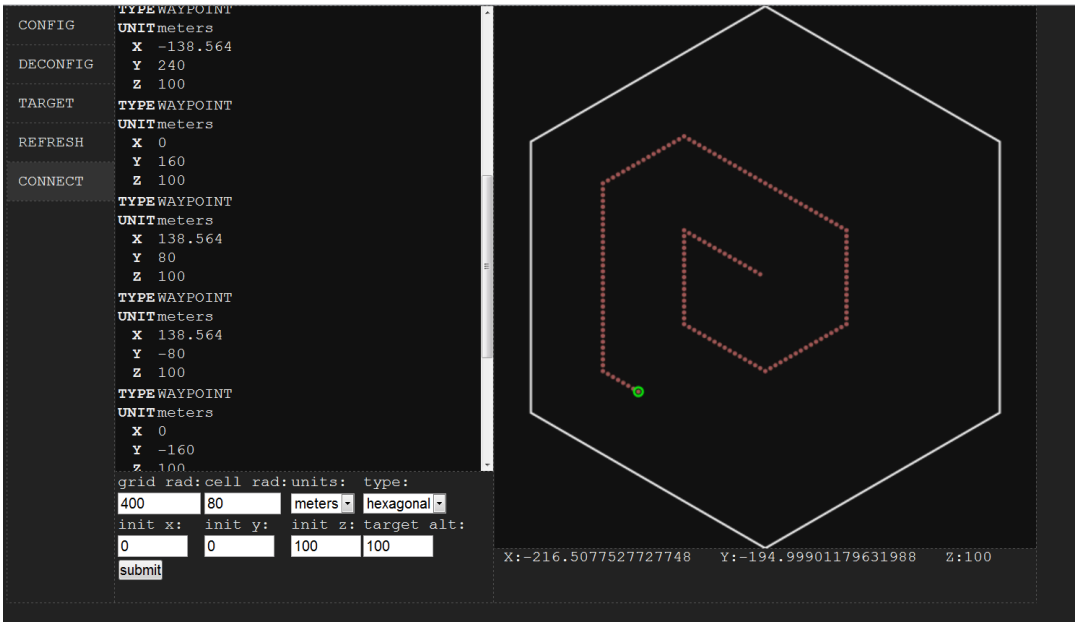


Figure 100: Software integration team's GUI used to view the path of the UAV during a search and rescue mission.

Figure 100 shows an image taken from the software integration's control user interface for their search and rescue waypoint navigation routine. The waypoints inserted into the simulation was generated from the output of their path planning software.

6.2 Camera

During the final flight of the Telemaster, the camera gimbal was setup with a Logitech web camera, which was controlled by the software team's image acquisition system. After approximately a minute and a half of image collection the software stopped collecting images, which was before takeoff. The cause of the pause in image collection was caused by USB problems inherent to the use of the linux distribution on their board. More details on the camera is discussed in the camera gimbal section of the platform development chapter of the report

7. Results

We created a list of goals for this project at the very beginning. These goals are summarized in Table 10, the right hand column summarizes how well we accomplished these goals during this project. For more detailed results on this project see the results sections imbedded in the Platform and Autopilot sections. Those sections cover detailed results on the simulations, gimbal performance, flight tests, and power board.

Table 10: Summarizes the how successfully we accomplished all of our Goals during this project.

Three fully autonomous fixed wing aircraft	
Interface with the UAV controlling search algorithm	
Autonomous navigation based on GPS waypoints	
Each UAV will need to carry a payload of 10 lbs	
Capable of flying at a cruising speed of 35 – 45 miles per hour	
Capable of flying between 120 ft. and 400 ft. of altitude as specified by the FAA for UAVs	
Capable of maintaining flight without refueling for a minimum of 1 hour	
Capable of supplying 102 watts of power to on board electronics, for a minimum 1 hour per charge	
Contains a pan tilts camera gimbal capable of holding up to two cameras, one color camera and one infrared camera.	

We only got half a smiley on having three fully autonomous fixed winged aircrafts because we were only able to get one flying under assisted flight, and one under RC flight, so we didn't fully meet our goals. We were unable to have any of them fly under Auto 2 because Robin crashed and we didn't have time to repair him before the project ended.

We gave our selves a full frown face on our third objective because we never got Auto2 to work on the real planes, only simulation, which was in charge of way point navigation. However in Simulation we were able to successfully tune the control loops and navigate between waypoints given from a Python Script, similar to how the onboard autopilot will be receiving waypoints from the software team's higher level board. A complete result on the Simulation is in the Autopilot results section.

We also only gave ourselves a half smiley on being capable of supplying 102 watts of power to the on board electronics because while we fully designed power board capable of supplying the necessary voltage it was never printed due to time and budget constraints.

The camera gimbal was not fully utilized because the software team was unable to get the Sony block camera working with their systems. So we didn't fully install the pan tilt system.

8. Conclusion and Recommendations

Project WiND has made great strides towards the ultimate goal of creating a multi – UAV system for use in search and rescue. The UAV Robin has successfully flown under AUTO1, however, suffered serious damage during a crash on April 11, 2012, and therefore is currently grounded indefinitely as a result. Jay has flown under radio control, and Duck is a few parts short of an RC flight. The autopilot boards are successfully connecting and communicating to a laptop over the Xbee radios, and multiple autopilot boards can connect to the same laptop simultaneously and stream data to the ground control software console on the laptop. We can also simulate multiple UAVs on different flight paths in the same console.

8.1 Recommendations for Future Development

In the future, this project needs to install the autopilot in Jay and successfully make AUTO 1 and AUTO 2 flights, ensuring that it can successfully navigate two or more preprogrammed waypoints. Duck needs to be finished and tested from start to finish. Robin needs the most amount of work due to the crash, which involves a new airframe. Once all of the UAVs are successfully flying under autopilot, the next step would be to integrate completely with the other teams.

Integration with the other projects would allow the users to fly the fleet of drones from the Microsoft Surface, update waypoints, and stream filtered images back from the different UAVs. To accomplish this, the software team needs to have an interface with the autopilot board in order to change waypoints as well as get current telemetry and location data. This process has been started, but an elegant solution has not been developed yet. Mechanically, the teams need to integrate all of the necessary hardware for each team into the UAVs, including the search and rescue cameras, FPGA board for image processing, the wifi modules, the power distribution boards, and the software team's logic board. All of these components need to be mounted in the UAVs with a vibration resistant mounting system and wired together. The layout inside the UAVs has been complete for all of these components with the help of CAD models of the airframes to ensure that each airframe remains balanced with the addition of all the extra weight.

Once these have been complete, the system will be fully functional and ready to perform low altitude, efficient aerial search and rescue.

9. Acknowledgments

The platform team of project wind would like to give thanks to the following people and organizations.

- Professor Padir
- Professor Wyglinski
- Professor Looft
- Professor Stafford
- Professor Hall
- Mr Richard Gammond
- Tanner Hiller Airport
- The Mathworks
- WPI Robotics Department
- WPI ECE Department
- Joseph Funk Sr.
- Project wind software team
- Project wind communications team

10. Authorship

1. Introduction: Written by Christopher Whipple, Joseph Funk
2. Platform Development: Written by Catherine Coleman, Joseph Funk
3. UAV Control Design: Written by James Salvati
4. Timeline: Written by Christopher Whipple
5. Budget: Written by Christopher Whipple
6. Integration with software and communication: Written by James Salvati
7. Results: Written by Catherine Coleman, James Salvati, Christopher Whipple, Joseph Funk
8. Conclusion and Recommendations: Written by Christopher Whipple

Appendix A: Written by Christopher Whipple
Appendix B: Written by Christopher Whipple
Appendix E: Written by Catherine Coleman, James Salvati, Christopher Whipple, Joseph Funk
Appendix F: Written by Catherine Coleman, Joseph Funk
Appendix G: Written by James Salvati
Appendix H: Written by Joseph Funk

11. Bibliography

Search and Rescue:

National Search and Rescue Manual. Volume 1: National Search and Rescue System.[United States]: Joint Chiefs of Staff Washington Dc, 1991. http://www.towage-salvage.com/files/sar01/.

Heggie, Travis W, Amundson, Micheal E. "Dead Men Walking: Search and Rescue in US National Parks". Wilderness & Environmental Medicine, vol. 20(3), pp. 244 – 249, 2009

Iserson, K V. "Injuries to Search and Rescue Volunteers. A 30-year Experience" The Western journal of medicine, vol. 151, Iss. 3, pp. 352 – 353 09/1989

UAV Current Uses:

Air Force Public Affairs, (January 5, 2012), "MQ-9 Reaper Fact Sheet" [online], Available: http://www.af.mil/information/factsheets/factsheet.asp?fsID=6405

Bolkcom, Christopher, "Homeland Security: Unmanned Aerial Vehicles and Border Surveillance" Congressional Research Service,12/2004

El Paso County Search and Rescue (2011), [online] Available: http://www.epcsar.org/

Eric Schmitt, (January 10, 2012), "3 killed as Drone Strikes Resume in Pakistan" New York Times [online] Available: http://www.nytimes.com/2012/01/11/world/asia/cia-drone-strikes-resume-in-pakistan.html?_r=1

Michael A. Goodrich, Bryan S. Morse, Damon Gerhardt, Joseph L. Cooper, Morgan Quigley, Julie A. Adams, Curtis Humphrey, "Supporting wilderness search and rescue using a camera-equipped mini UAV" Journal of Field Robotics, Vol. 25, Iss. 1-2 pg 89-110, 2008

Russell Naughton.(June 22, 2003). The 'Aerial Target' and 'Aerial Torpedo' in the USA (first)[Online].Available: http://www.ctie.monash.edu/hargrave/rpav_usa.html

Airplane research:

Bill Smith (2006). Blue Foam Wing Construction (first) [Online].Available: http://www.clapa.org/BLUE%20FOAM%20CONSTRUCTION%20FOR%20CONTROL%20LINE.pdf

Ron Alexander (October, 1997). Building Composite Aircraft (first)[Online].Available: http://www.sportair.com/articles/Building%20A%20Composite%20Aircraft.html

Mike Harris.(2005).Weight Analysis of Composite Airframe Construction (first)[Online].Available: http://www.privatedata.com/byb/rocketry/composites/composite_airframes.html

Ron Alexander.(May 1999).Basics of Composite Construction(first)[Online].Available: http://exp-aircraft.com/library/alexande/composit.html

Honus.(June 3, 1997). Simple Methods for Molding Fiberglass and CarbonFiber (first)[Online].Available: http://www.instructables.com/id/Simple-methods-for-molding-fiberglass-and-carbon-f/

Hobby Lobby.(January, 2012). AXI Gold 4120/18 Out Runner Motor (first)[Online].Available: http://www.hobby-lobby.com/axi_gold_4120_18_outrunner_motor_3039_prd1.htm

Hobby Lobby.(January, 2012). 4500mAh 4S 14.8V 20C LiPo Battery w/ Deans (first)[Online].Available: http://www.hobby-lobby.com/4_cell_14.8v_4500mah_20c_lipo_pack_364091_prd1.htm

Arthur Golnik (2003).Energy Density of Gasoline (first)[Online].Available: http://hypertextbook.com/facts/2003/ArthurGolnik.shtml

Barnard Microsystems Limited .(2012).High Energy Light Weight Batteries(first)[Online].Available: http://www.barnardmicrosystems.com/L4E_batteries.htm

Isidor Buchmann.(2012).What's the Best Battery?(first)[Online].Available: http://batteryuniversity.com/learn/article/whats_the_best_battery

University of Alberta. (2011, September). Alberta UASGroup (first)[online] Available: http://paparazzi.enac.fr/wiki/UAlberta_UASGroup

Autopilot Research:

Images:

http://www.cloudcaptech.com/piccolo_II.shtm
http://diydrones.com/page/autopilots-1
http://paparazzi.enac.fr/wiki/File:Ir_response_curve.gif

Sources:

Am Cho; Jihoon Kim; Sanghyo Lee; Sujin Choi; Boram Lee; Bosung Kim; Noha Park; Dongkeon Kim; Changdon Kee.(2007 Oct.). "Fully automatic taxiing, takeoff and landing of a UAV using a single-antenna GPS receiver only," Control, Automation and Systems, 2007. ICCAS '07. International Conference on. [Online], vol., no., pp.821-825, 17-20 doi: 10.1109/ICCAS.2007.4407014 Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4407014&isnumber=4406493

Abdelkrim, N.; Aouf, N.; Tsourdos, A.; White, B. (2008 June) "Robust nonlinear filtering for INS/GPS UAV localization," Control and Automation, 2008 16th Mediterranean Conference. [online] vol., no., pp.695-702, 25-27
doi: 10.1109/MED.2008.4602149

Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4602149&isnumber=4601965

Almeida, Pedro.; Bencatel, Ricardo.; Goncalves, Gil.; Sousa, Joao. (2006). "Multi-UAV Integration for Coordinated Missions" [online]
Available : http://whale.fe.up.pt/asasf/images/f/fa/Multi-UAV_Integration_for_coordinated_missions.pdf

Almeida, Pedro.; Bencatel, Ricardo.; Goncalves, Gil M.; Sousa, Joao Borges.; Ruetz, Christoph, Ruetz. (2007) "Experimental Results on Command and Control of Unmanned Air Vehicle Systems
Available: http://paginas.fe.up.pt/~dee07011/uploads/Main/UAV_Command_Control.pdf

Bayraktar, S.; Fainekos, G.E.; Pappas, G.J. (2004, December) "Experimental cooperative control of fixed-wing unmanned aerial vehicles," Decision and Control, 2004. CDC. 43rd IEEE Conference [online] , vol.4, no., pp. 4292- 4298 Vol.4, 14-17
doi: 10.1109/CDC.2004.1429426
Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1429426&isnumber=30838

Beard, Randal.; Kingston, Derek.; Quigley,Morgan.; Snyder, Deryl.; Christiansen, Reed.;Johnson, Walt.;McLain Timothy.; Goodrich, Michael. (2005, January). "Autonomous Vehicle Technologies for Small Fixed-Wing UAVs" Journal of Aerospace Computing, Information, and Communication Vol. 2,
Available: http://morse.colorado.edu/~timxb/Aiaa.pdf

Beyeler, Antoine.; Zufferey, Jean-Christophe.; Floreano, Dario. (2009) "optiPilot: control of take-off and landing using optic flow" European Micro Air Vehicle conference and competition. [online]
Available: http://infoscience.epfl.ch/record/140575/files/emav09_optipilot_web.pdf

Bin, He.; Justice, Amahah. (2009, March) "The design of an unmanned aerial vehicle based on the ArduPilot" Indian Journal of Science and Technology. Vol. 2, No. 4.
Available: http://www.indjst.org/archive/vol.2.issue.4/apr09justice.pdf

Bronz, Murat.; Moschetta, Jean Marc.; Brisset Pascal.; Gorraz, Michel. (2009, December) "Torwards a Long Endurance MAC" International Journal of Micro Air Vehicles, vol. 1, nu 4, pp 241-254. [online]
Available: http://multi-science.metapress.com/content/qx60752m05k45439/

C. B. Low, "A Trajectory Tracking Control Design for Fixed-wing Unmaned Aerial Vehicles," in 2010 IEEE Internal Conf on Control Applications., Yokohama

Chang-Sun Yoo; Iee-Ki Ahn; (2003 Oct.), "Low cost GPS/INS sensor fusion system for UAV navigation," Digital Avionics Systems Conference, 2003. DASC '03. The 22nd , [online] vol.2, no., pp. 8.A.1- 8.1-9 vol.2, 12-16
doi: 10.1109/DASC.2003.1245891
Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1245891&isnumber=27920

Chao, HaiYang.;Cao, YongCan.;Chen, YangQuan. (2010, February) “Autopilots for small unmanned aerial vehicles: A survey” International Journal of Control, Automation and Systems [online] Vol 8, no.1,pp36-44, February 2010 DOI: 10.1007/s12555-010-0105-z
Available: http://www.springerlink.com/content/m2x11674184m208x/

Dobrokhodov, V.N.; Kaminer, I.I.; Jones, K.D.; Ghabcheloo, R.. (2006, June) "Vision-based tracking and motion estimation for moving targets using small UAVs," American Control Conference, 2006. [online] vol., no., pp.6 pp., 14-16
doi: 10.1109/ACC.2006.1656418
Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1656418&isnumber=34689

Euston, M.; Coote, P.; Mahony, R.; Jonghyuk Kim; Hamel, T. (2008, September) "A complementary filter for attitude estimation of a fixed-wing UAV," Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference.[online] vol., no., pp.340-345, 22-26 Sept. 2008
doi: 10.1109/IROS.2008.4650766
Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4650766&isnumber=4650570

Garcia, Esteban Gonzalez.; Becker, John. (2007) “UAV stability derivatives estimation for hardware-in-the-loop simulation of Piccolo autopilot by qualitative flight testing” 1st Latin American UAV Conference. [online]
Available: http://www.aerodreams-uav.com/docs/aeroduav-conf.pdf

Jensen, A.M.; Baumann, M.; YangQuan Chen. (2008, July) "Low-Cost Multispectral Aerial Imaging using Autonomous Runway-Free Small Flying Wing Vehicles," Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International. [online]. vol.5, no., pp.V-506-V-509, 7-11
doi: 10.1109/IGARSS.2008.4780140
Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4780140&isnumber=4780001

Joseph, Lum Yue Hao. (2011) “Unmanned Aerial Vehicle Autopilot System”. [online]
Available: http://dynamicslab.mpe.nus.edu.sg/dynamics/thesis1011/UAV%20Autopilot.pdf

Jung, D.; Levy, E.J.; Zhou, D.; Fink, R.; Moshe, J.; Earl, A.; Tsiotras, P.;(2005 Dec.) , "Design and Development of a Low-Cost Test-Bed for Undergraduate Education in UAVs," Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference [online] , vol., no., pp. 2739- 2744, 12-15
doi: 10.1109/CDC.2005.1582577
Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1582577&isnumber=33412

Kabbabe, Kristopher. (2011) “Development of Procedures for Flight Testing UAVs using the ArduPilot System”. [online]
Available: http://uavs.mace.manchester.ac.uk/uploads/Research/MScKabbabe2011.pdf

Kim, Jong-Hyuk.; Sukkarieh, Salah.; Wishart, Stuart. (2006) “Real-Time Navigation, Guidance, and Control of a UAV Using Low-Cost Sensors” Springer Tracts in Advance Robotics [online] vol.24, pp 299-309, 2006, DOI: 10.1007/10991459_29
Available: http://www.springerlink.com/content/h548798460354551/

Metni, Najib.; Pflimlin, Jean-Michel.; Hamel, Tarek.; Soueres, Philippe. (2006, December) “Attitude and gyro bias estimation for a VTOL UAV” Control Engineering Practice. [online] vol. 14, is. 12, pp 1511-1520.
Available: http://www.sciencedirect.com/science/article/pii/S096706610600030X

Reuder, Joachim.; Brisset, Pascal.; Jonassen, Marius.; Müller, Martin.; Mayer, Stephanie. (2009, April) “The Small Unmanned Meteorological Observer SUMO: A new tool for atmospheric boundary layer research” Meteorologische Zeitschrift, vol. 18, nu 2, pp. 141-147.
Available:http://www.ingentaconnect.com.ezproxy.wpi.edu/content/schweiz/mz/2009/00000018/00000002/art00004

Zhihai He; Iyer, R.V.; Chandler, P.R. (2006, June). "Vision-based UAV flight control and obstacle avoidance," American Control Conference, 2006 . [online] vol., no., pp.5 pp., 14-16
doi: 10.1109/ACC.2006.1656540
Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1656540&isnumber=34689

Appendix

Contents

APPENDIX A: SAFETY MANUAL OF PROJECT WIND 100
APPENDIX B. PROJECT WIND INCIDENT REPORT 108
APPENDIX C. PROJECT WIND LIABILITY RELEASE FORM 109
APPENDIX D: MASTER BUDGET LIST 113
APPENDIX E: DETAILED FLIGHT LOG 114
APPENDIX F: USER MANUAL 116
APPENDIX G: POWER BOARD SCHEMATIC 132
APPENDIX H: LM5118 DATA SHEET 141

Appendix A: Safety Manual of Project Wind

The Safety Manual for Project WiND

In Case of Emergency: **Call 911**

Safety Protocol

Project WiND is dedicated to safety. This manual has been developed to ensure the safe usage of any products produced or used by project WiND under both test circumstances as well as real world applications. Project WiND will be using unmanned aerial vehicles (UAVs), which are dangerous, and therefore must be treated as such. What makes these UAV's so dangerous are their size and their power. The smallest UAV will have a 6 foot wing span, while the largest may have upwards of a 12 foot wingspan. They are also powered by either an electric motor or a gas engine, either of which, when attached to a propeller, have enough torque to remove a finger from your hand. These UAVs will be controlled by computers, not humans, which increase the risk of an accident during testing and operation. However, if the proper precautions are taken, project WiND can ensure the safety for all involved. These safety precautions are outlined here in this manual.

All members of the team must read this manual in its entirety and understand all of these protocols before any of the robots leave the ground.

Definitions

Unmanned Aerial Vehicle

An unmanned aerial vehicle, abbreviated to UAV, is an aircraft which is under the direct control of a computer for navigation and stability control. The computer controls all movement of the aircraft, and monitors all systems using sensors to ensure stable flight. UAVs are capable of flying from GPS coordinate to GPS coordinate for navigation. For the purposes of this document, the term "UAV" and "robot" are to be used interchangeably.

Academy for Model Aeronautics

The Academy for Model Aeronautics, abbreviated to AMA, is a nationwide governing body for the safe use of radio controlled (RC) aircraft, including both planes and helicopters. AMA offers certification to prospective RC pilots. Members have access to fly at AMA flying fields around the country. Certification also offers the certified member some insurance to help with expenses in the case of property damage due to a crash. See Appendix D for more details on the benefits of AMA and their rules and regulations. Project WiND will follow all of the AMA safety regulations, in addition to the regulations specified in this document.

Definition of Robot States

During the test, each robot will go through a series of states, defined here.

Name	Systems Active	People allowed near robot	Notes
Cold	None	Anyone	Used for Maintenance and inspection
Pre-warm	Transmitter On	Anyone	MUST pass through Pre-warm from either warm or cold
Warm	Transmitter On, Electrical Control System active	Project WiND Personnel	Engine Remains off
Hot	Engine On, All electronics are powered	Flight Controller, explicitly invited WiND personnel	This ONLY happens outdoors at test sites, with the intention of flying.

Hot and Safe Zones

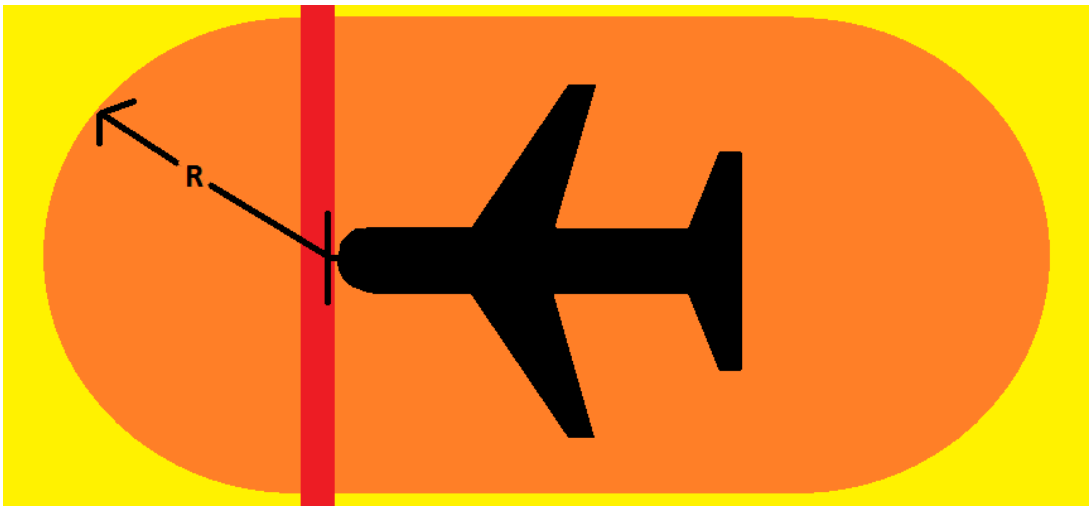
Once on location at a test site, the area being used for testing must be divided up to give safe zones and danger zones for WiND personnel as well as spectators. AMA fields will have similar zones already in place and marked. In the event that the test is being conducted outside an AMA field, here is a description of how to define these zones.

Defining the Hot Zone

The size of the hot zone is specified by the amount of runway needed by the robot. Each robot is a different size, and will require different amounts of take off and landing space. The hot zone must be physically marked using cones, tape etc. The area needs to be flat, continuously open space. The type of terrain the runway is depends on how the robot landing gear is configured. For example, if the landing gear is made for soft ground, then a grass runway will suffice. If there is snow on the ground, there will have to be the potential to either clear a runway of snow OR apply snow skis to the robot to allow it to safely land on snow covered runways. AMA (Academy of Model Aeronautics) fields will have this area marked already.

Hot Zone around the Robot

While on the ground, the robots will always remain in the hot zone. Figure 0.1 is a picture of an airplane with the area immediately around the plane designated with colors. A description of the colors is below.



0.1 Immediate Robot Danger Zones (NOT to scale)

Red: The area in line with the propeller. This is an **immediate danger zone**, **stay out.**

Orange: The area immediately around the robot. The size of this area is to be calculated using the length of the wings. The Value labeled 'R' in the figure is the wingspan of the robot. Only WiND personnel and their engine experts working on getting the robot flight ready are allowed inside the orange zone. No other robots or foreign debris are allowed in the orange.

Yellow: The rest of the Hot Zone as defined above. The rules for the hot zone apply.

Defining the Safe Zone

Once the runway has been defined, the Flight Controller in coordination with the ground controller needs to define the safe zone. This is where all personnel and any spectators with the exception of the flight controller will stand while the robot is declared hot (See declaration of Robot stages for explanation) or in the air. It is important to think about the direction of flight intended for the robots. If they are headed

to points north of the runway (hot zone), then the safe zone should be to the south of the hot zone in order to minimize robots flying over the safe zone. AMA (Academy of Model Aeronautics) fields will have this area marked already.

The Safety Positions for Single Robot Testing:

1). **The Flight Controller.** Whenever a robot leaves the ground, a flight controller must be standing by with an RC override switch on an RC module prepared to take control from the auto pilot in the event of an in flight emergency. The flight controller is the lead safety person on the field at all times. What he / she says goes for everyone involved. The flight controller is required to:

A) Be a card carrying member of AMA (Academy of Model Aeronautics)

B) Lead project WiND in determining a good location for tests as well as defining a flight plan, as defined in the section "Explanation of flight plans"

C) Define a safe zone and a hot zone, based off where the robots will take off and land. Other than the ground controller, NO Person under ANY circumstances is allowed into the hot zone without the consent of the flight controller. This is to be done before the robot flies.

D) Hold a safety briefing with all personnel and spectators, ensure that everyone understands the rules outlined by AMA as well as those outlined in this manual and sign any liability forms necessary.

E) Keep eye contact with the robot at all times while taxiing on the ground or in flight. The flight controller is to be vigilant, constantly assessing where the robot is, and where the robot is going to be in the immediate future. They need to be watchful of other planes, trees, buildings, power lines and birds. Even if the robot is under the control of the auto pilot, the flight controller needs to be ready to override the auto pilot and switch to RC mode.

F) Announce a landing to all personnel and spectators with at least 2 minutes notice in the case of a scheduled landing and as soon as possible in the event of an unscheduled or emergency landing. The announcement should declare "Landing" plus an estimated time of landing, for example, "Two minutes to landing".

G) On the robot's final approach, the Flight controller is to make one final announcement of "Final approach" to all personnel, and verify with the ground controller that the runway is clear and safe for landing. After this announcement, the flight controller is free to land the plane.

2). **The Ground Controller**. The ground controller will be the second in the safety hierarchy. He or she will predominately control any ground based movements of both people and robots. Their specific responsibilities include:

A) Defining the hot zone in coordination with the Flight controller. See defining the hot zone for details.

B) Ensuring that all personnel and spectators abide by the hot zone and safe zone rules. They must be vigilant in watching for personnel movement in the hot zone and warn any unsuspecting spectators (anyone in the immediate area not affiliated with the project) about the robot movements.

C) Ensure that the hot zone remains clear of people, debris, animals and other robots while there are robots airborne.

D) Coordinate the landing of the robot, and certify for the flight controller that the runway is clear for landing.

E) Communicating with airport personnel in the event that a test is to be conducted at an airport or AMA field.

F) The ground controller is the ONLY person allowed to talk to the flight controller. They are to relay any messages from the flight controller (who is consistently primarily focused on the robot movements) to the other personnel and spectators, and they will filter any communications from personnel and spectators to the Flight controllers. Since the flight controller needs to remain focused on the robot, this communication needs to be at an absolute minimum, which makes this role of the ground controller very important. Acceptable communications include comments about visible hardware malfunctions, like broken landing gear or visible smoke, etc.

The Safety Positions for Multiple Simultaneous Robot Testing:

Testing multiple robots at once will be a difficult task. Much more communication and organization is necessary to make the test successful, therefore there are a few additional safety positions for just such circumstances.

1). **The Chief Flight Controller**. The chief flight controller is a flight controller, and has all of the responsibilities of a flight controller as specified above. In addition to everything stated above, the Chief flight controller is responsible for:

A) Holding a flight controllers meeting before the robots are even brought out onto the field. At these meetings, the chief flight controller is to review any maneuvers to be undertaken by the test that day. He or she needs to determine what exactly the robots will be doing and when. See “Explanation of Flight Plans” for details.

B) Coordinating multiple robots in the air and on the ground by communicating with the other flight controllers for each robot in the test.

2). **The Ground Controller**. The ground controller takes the same role as when there is only one robot being operated. He or She is still second in the safety hierarchy. All responsibilities are the same.

3).**The Spotter**. The spotter is a specific role for when there are multiple robots in operation. He or she is the big picture safety person. It is the responsibility of the spotter to Watch the robots as they fly and in particular watch for impending collisions between planes and object on the ground, like trees, buildings and people. In the event of an impending collision, please inform the communications runner to avert said collision.

4). **The Communications Runner**. The communications runner is a specific role for when multiple robots are in operation. It is the responsibility of the communications runner to ensure accurate communication between the different flight controllers, the spotter and the ground controller.

5). Each additional **flight controller** (one for each additional robot in the test over 1), because each robot has to have a unique flight controller, each with its own radio transmitter being controlled by the individual flight controllers. These additional flight controllers have equal stature in the hierarchy, and need to take commands from both the ground controller and the chief flight controller.

Procedural Safety

Every time a robot is expected to fly, these are the steps to be taken to ensure safe operation of the robots for WiND personnel as well as spectators.

1). Location Selection

The first consideration to take is to understand the size of both the robot being flown as well as the size of the flight path to be flown. The location of where the flights will take place must be determined before moving to step 2. Considerations for the selection of a location should include:

A) The scale of the flights. If the scale is to be on the order of thousands of feet, an AMA field will suffice.

B) If the scale of the tests being conducted is in the scale of miles, a large enough area of uninhabited land or airport will be required in order conduct the tests.

For example, if the test being conducted is using a single eight foot wingspan RC drone, the test field should be a standard AMA flying field. If the test consists of multiple communicating UAV's with 8+ foot wingspans, the test area will need to be approximately 3+ square miles of uninhabited space.

2). Explanation of flight plans

As far as the actual flight plans go, it is important for any and all of the flight controllers to understand exactly what the robots are going to be doing, not only on paper but also spatially in the real world. Things to keep in mind include:

A) Avoid flying over spectators and flight controllers. Set up the flight plans to travel away from the safe zone

B) Ensure line of sight between were the robot is expected to go and where the flight controllers will be standing.

C) Avoid trees

D) Avoid power lines by 5000 feet at all times. If power lines are even present at the field, reconsider the location.

How high and how fast the robots can fly are to be dictated by the FAA and AMA where applicable. In general, we need to keep the robots less than 400 feet off the ground. Speed should not be an issue due to our heavy payload and intentionally slow propeller selection. See Appendix D for more information from AMA and their rules and regulations.

3). Transportation of the Robot

Due to the nature of the project, it is inevitable that the robots will have to be transported from the build facility to a safe place to fly (See description of flight plans and definition of Hot Zone for details). When transporting the robot, follow these guidelines to ensure the safety of the WiND personnel as well as the safety of the robot.

A) No fuel or batteries are to be in, on, or around the robot during transportation.

B) Fuel will be carried in a gasoline safe container in a different vehicle from the robot. The container is to be both DOT and UL certified, and is to never be less than half full, and never more than 95% full to allow for expansion.

C) The robot is required to fit entirely into the vehicle in which it is to be transported. For example, strapping the robot to the top of a passenger car is unacceptable, as is having the robot hang off the back of a pickup truck.

D) The robot, depending on size, may need to be transported in several pieces and rebuilt in the hot zone of the test site.

4). Liability Forms

Any and all team WiND personnel, advisors and any spectators that will be present for a project WiND test flight must read and sign a liability release form from WPI. This form, specialized for Project WiND, can be found as Appendix C in this document. This is to be done before or upon arrival to the test site before the robots are unloaded from the transportation vehicle. It is the responsibility of the fight controller or the chief flight controller (when applicable) to ensure all spectators and personnel have signed this form.

5). The Pre flight Check

Before a robot leaves the ground, the robot **must** pass a thorough inspection by the project WiND team. All testing is to be done while the robot is in the warm state (see definition of robot states). This is to be done before every takeoff, even if the robot flew earlier that day. The inspection includes testing:

A) All flight control surfaces, both by the manual override transmitter, as well as a safety routine programmed into the autopilot (if the auto pilot is going to be used). This is to be done while the robot is in the warm state (see definition of robot states).

B) The manual override between the auto pilot and the transmitter. Ensure all flight control operations seamlessly transition from one to the other

C) Inspect the airframe for damage

D) Test the communications links between the multiple robots as well as the communication between the auto pilot and the high level board.

E) Test the sensors and ensure accurate readings. **Be sure to zero any measurements**.

In the event that something is discovered to be damaged or malfunctioning during the pre flight check, it is up to the appropriate project WiND team to fix the issue before the robot will be permitted to fly. For example, if the team discovers damage to the landing gear, the hardware team

will be responsible for fixing and repairing this before the robot can take off. If the radio used to communicate with the other robots is malfunctioning, the communications team will need to solve the issue before a flight can proceed. **Each robot involved in a test MUST complete EACH AND EVERY point listed above to be qualified to fly in the test.**

6). Starting the Engine

These robots will be powered by a primary gas engine with a direct link to the propeller. This is a very dangerous set up if not handled properly. Project WiND is in contact with gas engine experts both from WPI and abroad to ensure the safe handling, and starting of these engines. These experts will be giving safety meetings to go over any and all precautions necessary for safely starting these engines. All personnel from the hardware team will be required to attend these meetings, however, any other project WiND members or advisors can also attend. Only WiND personnel with this training are allowed to be near the engine while it is fueled, getting ready to be started, or running. While at test sites, only the hardware team is allowed to attempt to start the engines, with permission from the robot's respective flight controller. For more information about where the hardware team is allowed to stand during startup, see definition of the 'Hot Zone around the Robot' and figure 0.1 above.

Other Procedures

Emergency Landing Protocol

There are two different scenarios for emergency landings. One is an unscheduled landing, which is defined as any time the auto pilot faults out and the robot is taken over by the human flight controller, and is brought down by for inspection. The urgency to land in this scenario is not pressing, and it is advised for the flight controller to regain full control of the robot and all the aircraft functions, announce the landing to the ground controller and any other flight controllers, and then proceed to carefully bring the robot in for a landing. The timing for such an event is to be scaled in minutes.

An emergency landing is defined as a loss in a major flight control system, such as the loss of control of the rudder or ailerons, etc and subsequent loss in the auto pilot. In this case, the urgency to bring the robot down safely is at its highest. The flight controller is to take control of the robot, and land in any area that they deem fit, even if this is outside of the hot zone. The first priority is to avoid people. The second priority is people's property, so avoid homes and buildings, as well as cars that may be parked nearby. The third is to bring the robot down with as little additional damage as possible. This process should take place in the tens of seconds.

The event of an emergency landing is the only time the safety hierarchy is to be altered. In this event, the flight controller controlling the robot in question assumes the position of chief flight controller and can direct the other robots out of the way to make for as safe of a landing as possible.

Indoor flight control surface testing

Testing the different systems indoors is permitted; however the robot is not allowed to pass beyond the warm state. The engines will never be turned over under any circumstances indoors. This allows testing on any of the electronics, flight control surfaces, and sensors in a controlled lab environment, without endangering any of the project WiND personnel.

Storage of the Robot

The robots are just as dangerous when they are in storage, mostly due to the fact that when the robot is being stored, there are no project WiND personnel attending it. These rules must be followed to ensure safe storage of the robot.

A) Keep the robot stored in a safe locked, and secured location, preferably by a key lock, a swipe RFiD card reader, or another comparable locking mechanism

B) Ensure the robot is stored without any batteries inside

C) Ensure any and all of the fuel has been drained

D) All access to the robot while in storage must go through the platform team or the overall team leader.

Adhering to these safety guidelines will ensure safe storage of the robot.

Reporting Incidents

In order to ensure continued safety, project WiND has created an Incident report form. This form, found as Appendix B of this document, is to be filled out in its entirety and returned to the team captain, the team advisors, as well as WPI's safety coordinator, in compliance with the WPI safety regulations. If the incident was an avoidable accident, then the safety manual shall be edited and revised to help prevent the incident from occurring again.

Remember, Project WiND has specified these outlines with your safety in mind; however it is your responsibility to abide by these rules and ensure safe robot operation for all!

I, _____ (your name, printed), hereby agree and certify that I have read and understood all the material covered in The Project WiND safety manual and agree to abide by all rules and regulations as specified by project WiND.

Signature: _____
Date:_____

Appendix B. Project WiND Incident Report

This is the official project WiND Incident report. Please fill out all of the fields and submit a copy of the completed form to:

Your team leader: Name_____
Project WiND Student Lead: Name_____

Faculty Advisor(s): Name_____
Name_____

WPI's Safety Coordinator: Name_____

Date: _____
Location of Incident:_____

Person(s) Involved: Name_____
Name_____
Name_____

Explanation of Events:

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Your Name (Printed): _____

Your Signature: _____ Date:_____

Appendix C. Project WiND Liability Release Form

WORCESTER POLYTECHNIC INSTITUTE

LIABILITY RELEASE FORM

SPECIAL ACTIVITY:

Project WiND Test Flight

The undersigned _____ "participant" does release and shall indemnify and hold harmless WPI, its officers, trustees, employees, and agents from and against all claims, damages, losses, and expenses including, but not limited to, medical expenses, attorneys fees, and court awards arising out of or resulting from any injury, sickness, disease or death occurring in connection with my participation in the prject WiND test flight.

Signature of this form verifies that the participant understands and confirms that he/she is volunteering to participate in the aforementioned activity at his/her own risk.

Signature of this form verifies that the participant is aware of and understands the potential, inherent dangers and risks involved in participating in this activity.

Signature of this form also verifies that the participant is covered by appropriate medical insurance for injuries or illnesses and further understands that any deductible, co-payments and uncovered claims will be the sole responsibility of the participant. This Liability Release shall be governed by Massachusetts law.

Participant's Signature **Date**

Appendix D: AMA Regulations and Documents

Academy of Model Aeronautics National Model Aircraft Safety Code

Effective January 1, 2011

A. GENERAL: A model aircraft is a non-human-carrying aircraft capable of sustained flight in the atmosphere. It may not exceed limitations of this code and is intended exclusively for sport, recreation and/or competition. All model flights must be conducted in accordance with this safety code and any additional rules specific to the flying site.

1. Model aircraft will not be flown:
   (a) In a careless or reckless manner.
   (b) At a location where model aircraft activities are prohibited.
2. Model aircraft pilots will:
   (a) Yield the right of way to all man carrying aircraft.
   (b) See and avoid all aircraft and a spotter must be used when appropriate. (AMA Document #540-D-See and Avoid Guidance.)
   (c) Not fly higher than approximately 400 feet above ground level within three (3) miles of an airport, without notifying the airport operator.
   (d) Not interfere with operations and traffic patterns at any airport, heliport or seaplane base except where there is a mixed use agreement.
   (e) Not exceed a takeoff weight, including fuel, of 55 pounds unless in compliance with the AMA Large Model Aircraft program. (AMA Document 520-A)
   (f) Ensure the aircraft is identified with the name and address or AMA number of the owner on the inside or affixed to the outside of the model aircraft. (This does not apply to model aircraft flown indoors).
   (g) Not operate aircraft with metal-blade propellers or with gaseous boosts except for helicopters operated under the provisions of AMA Document #555.
   (h) Not operate model aircraft while under the influence of alcohol or while using any drug which could adversely affect the pilot's ability to safely control the model.
   (i) Not operate model aircraft carrying pyrotechnic devices which explode or burn, or any device which propels a projectile or drops any object that creates a hazard to persons or property.
   Exceptions:
   - Free Flight fuses or devices that burn producing smoke and are securely attached to the model aircraft during flight.
   - Rocket motors (using solid propellant) up to a G-series size may be used provided they remain attached to the model during flight. Model rockets may be flown in accordance with the National Model Rocketry Safety Code but may not be launched from model aircraft.
   - Officially designated AMA Air Show Teams (AST) are authorized to use devices and practices as defined within the Team AMA Program Document (AMA Document #718).

   (j) Not operate a turbine-powered aircraft, unless in compliance with the AMA turbine regulations. (AMA Document #510-A).
3. Model aircraft will not be flown in AMA sanctioned events, air shows or model demonstrations unless:
   (a) The aircraft, control system and pilot skills have successfully demonstrated all maneuvers intended or anticipated prior to the specific event.
   (b) An inexperienced pilot is assisted by an experienced pilot.
4. When and where required by rule, helmets must be properly worn and fastened. They must be OSHA, DOT, ANSI, SNELL or NOCSAE approved or comply with comparable standards.

B. *RADIO CONTROL (RC)*

1. All pilots shall avoid flying directly over unprotected people, vessels, vehicles or structures and shall avoid endangerment of life and property of others.
2. A successful radio equipment ground-range check in accordance with manufacturer's recommendations will be completed before the first flight of a new or repaired model aircraft.
3. At all flying sites a safety line(s) must be established in front of which all flying takes place (AMA Document #706-Recommended Field Layout):
   (a) Only personnel associated with flying the model aircraft are allowed at or in front of the safety line.
   (b) At air shows or demonstrations, a straight safety line must be established.
   (c) An area away from the safety line must be maintained for spectators.
   (d) Intentional flying behind the safety line is prohibited.
4. RC model aircraft must use the radio-control frequencies currently allowed by the Federal Communications Commission (FCC). Only individuals properly licensed by the FCC are authorized to operate equipment on Amateur Band frequencies.
5. RC model aircraft will not operate within three (3) miles of any pre-existing flying site without a frequency-management agreement (AMA Documents #922-Testing for RF Interference; #923- Frequency Management Agreement)
6. With the exception of events flown under official AMA Competition Regulations, excluding takeoff and landing, no powered model may be flown outdoors closer than 25 feet to any individual, except for the pilot and the pilot's helper(s) located at the flight line.
7. Under no circumstances may a pilot or other person touch a model aircraft in flight while it is still under power, except to divert it from striking an individual. This does not apply to model aircraft flown indoors.
8. RC night flying requires a lighting system providing the pilot with a clear view of the model's attitude and orientation at all times.
9. The pilot of a RC model aircraft shall:
   (a) Maintain control during the entire flight, maintaining visual contact without enhancement other than by corrective lenses prescribed for the pilot.
   (b) Fly using the assistance of a camera or First-Person View (FPV) only in accordance with the procedures outlined in AMA Document #550.

C. FREE FLIGHT

1. Must be at least 100 feet downwind of spectators and automobile parking when the model aircraft is launched.
2. Launch area must be clear of all individuals except mechanics, officials, and other fliers.
3. An effective device will be used to extinguish any fuse on the model aircraft after the fuse has completed its function.

D. CONTROL LINE

1. The complete control system (including the safety thong where applicable) must have an inspection and pull test prior to flying.
2. The pull test will be in accordance with the current Competition Regulations for the applicable model aircraft category.
3. Model aircraft not fitting a specific category shall use those pull-test requirements as indicated for Control Line Precision Aerobatics.
4. The flying area must be clear of all utility wires or poles and a model aircraft will not be flown closer than 50 feet to any above-ground electric utility lines.
5. The flying area must be clear of all nonessential participants and spectators before the engine is started.

Academy of Model Aeronautics
5161 East Memorial Drive
Muncie, Indiana 47302
Claims Contact Information
(765) 287-1256 – Business
(765) 286-3303 – Fax
claims@modelaircraft.org

How to file an insurance claim

1. The AMA member causing the accident should call AMA Headquarters and provide an initial report of the incident. Serious injury and major property damage losses *must* be reported immediately. AMA will forward the appropriate claim form with specific instructions to the member, club officer, and/or Contest Director (if the accident occurred at a sanctioned event).

2. The AMA member returns the *completed* and signed form *and* any necessary documentation required for the claim. Upon receipt of the necessary documentation, AMA will forward the information to the appropriate independent claims adjuster/Third Party Administrators (TPA) who will review and process the claim.

NOTES:

AMA insurance is "secondary" or "excess" insurance. So in addition to notifying AMA Headquarters of the accident, this also means an AMA member *must* first file the incident through any other insurance available to him or her: i.e. homeowners, renters, automobile, health insurance, etc. If the primary insurance covers the total claim amount, the member should let the AMA know so that we may close the file.

Claims are handled in the order in which they are received. Members are asked to please be patient during the summer because this is our peak time for claims. Claims are processed as quickly as possible.

Deductibles: All deductibles are the responsibility of the insured member ($250 for property damage, no deductible for bodily injury, $750 for medical claims, and $100 for Fire, Theft, & Vandalism claims). This deductible is separate from any other deductibles involving primary coverage for other insurance companies (typically homeowners or group health through employers).

Final decisions on what is covered and how much is paid are made by an independent Third Party Administrator (TPA) based upon the terms and conditions of the appropriate policy.

AMA members receive up to three types of insurance coverage with their membership as a benefit:[1]

- *Accident/Medical (AD&D)*: this coverage applies when a member injures himself or herself in a model-related incident.
- *Commercial General Liability (CGL):* this coverage applies when a member damages someone else's property or injures someone else. Damage to a member's own property is *not* covered since you cannot be liable to yourself, and there is *no* coverage if a member injures someone in his own family.
- *Fire, Vandalism, and Theft (FTV):* this coverage applies when a member's aircraft and/or RC equipment is damaged, destroyed, or stolen. Theft coverage applies *only* when the theft is from a member's locked vehicle or dwelling (including member's garage).

Coverage under any of the above policies is subject to actual terms and conditions of the policy. A specimen copy is located in the AMA Documents section of the AMA website.

[1] Park Pilot members only receive liability coverage up to $500,000 as a benefit through their Park Pilot membership. AD&D and FTV are not included.

Academy of Model Aeronautics
5161 East Memorial Drive
Muncie, Indiana 47302
(765) 287-1256 – Business
(765) 289-4248 – Fax
(800) 435-9262 – Membership Services
http://www.modelaircraft.org

FLYING SITE SAFETY AND OPERATIONAL RULES

The *Official AMA National Model Aircraft Safety Code* is a basic document for ALL AMA chartered clubs. Every member should be familiar with the Safety Code, and new members should receive instruction about this code.

Each club should also develop a carefully structured set of rules for their club's flying site. These rules should cover two (2) areas:

1) **Safety** rules for actual flying, and
2) **Operational** rules which include items concerning times of operation, permits required, emergency numbers, etc.

Following is a generic sample of a set of rules designed to supplement the required current *Official AMA National Model Aircraft Safety Code*. This example is not intended to represent the specific needs of all flying sites. It is, rather, intended to be a guide to help each club develop rules which specifically meet the individual site situation.

Appendix E: Master Budget List

	Price	Quanty	total
Senior telemaster	$209.99	1	$209.99
Hitec HS-322 Servo	$9.99	5	$49.95
RCG 20 cc Gas engine	$159.99	1	$159.99
propellor 16x4	$2.39	2	$4.78
Opto Gas Kill Switch	$11.83	1	$11.83
9 Channel remote	$89.00	1	$89.00
Batt for Ign.	$15.00	1	$15.00
Charger	$16.00	1	$16.00
Re-fueling pump	$24.00	1	$24.00
Fuel Lines for pump	$5.00	1	$5.00
Starter	$25.00	1	$25.00
Propeller	$12.00	2	$24.00
Tank, 32oz	$10.50	1	$10.50
Re fueling ports	$16.69	2	$33.38
Fuel Line	$5.00	3	$15.00
Fuel Filter	$3.47	3	$10.41
106" Giant Scale 30-50CC Skyline Champ	$299.00	2	$598.00
RCG 30cc Gas engine	$223.00	2	$446.00
Hitec HS-322 Servo	$9.99	12	$119.88
Opto Gas Kill Switch	$11.83	2	$23.66
DX6i radio system	$209.99	1	$209.99
Quick Disconnect valves	$17.00	4	$68.00
Clear Domes	$30.00	1	$30.00
Power HD High-Torque Servos (gimbal)	$20.00	0	$0.00
Yapa2 Autopilot	$80.00	2	$160.00
MediaTek GPS	$38.00	2	$76.00
MPXV7002DP Airspeed	$25.00	1	$25.00
Thermopiles	$140.00	2	$280.00
Propellor 18x6	$15.29	2	$30.58
Total:			$2,770.94

Appendix F: Detailed Flight Log

Detailed Flight Log.

The next step towards getting a fully autonomous UAV was to create a stable platform. The first Airframe we flew was Robin (the red telemaster). We flew Robin on November 13th, and February 2nd, 9th, 15th, and 18th. These flights gave us valuable information about the condition of the airframe and its stability. After every flight we made necessary adjustments to the airframe to help it fly better.

November 13th flight

Accompanying us on our first flight was Mr. Gammon, our engine and RC expert. After a few short successful flights, we discovered that we had excessive right thrust that resulted in a pull to the right during full power climbs. We also found that it needed more down thrust because it had excessive climb in altitude at full power. We fixed this for our next flight by tilting the engine down two more degrees and two more degrees to the right. We also found that our linkage for the throttle was off and created a non linear response. The engine was also idling too high and running excessively rich on the high end of rpm, which was fixed by adjusting the mixing and idol screws on site. During our final flight that day the engine stalled, which we latter found was caused by a kink in the fuel line. Mr. Gammon also recommended that we pressurize the gas tank with the exhaust from the muffler for the next flight. Our rudder and elevator horns were also put on backwards for this flight, which was fixed for the next flight. Lastly, we determined that the current servo being used for the rudder was too slow, and replaced with a faster servo.

Due to inclement weather, we were unable to fly at all during the months of December and January, so our next flights occurred in February. During December and January, we made several additional modifications to the plane. We moved the gas tank back 4 inches to help with keeping a consistent center of gravity as the fuel weight decreases with time during a flight, which increased our overall flight stability. Moving the gas tank bank also allowed us to move the throttle servo to the front of the plane closer to the engine. In this new position the linkage still needed some work and was still giving a nonlinear response, making it difficult to adjust speed efficiently. We also put a nipple on the muffler to pressurize the gas tank. With these corrections we were ready for the weather to break and continue testing.

February 2th flight:

During this flight we were able to test the GPS down link and we flew Robin without refilling the gas tank for 30 min. However our joint of J-B weld epoxy holding the new nipple to the muffler for pressurizing the gas tank melted, leaving a hole in the muffler. This did not stop us from making more flights, however. During the next flight, an aggressive move maneuver caused the wings to shift enough to sheer off one of the connection point of the wing struts.

For future flights we reduced the throw on the control surfaces for gentler movements. Our fix for this was to make new piece for the wing struts to connect to the wings. To prevent the wings from shifting again, we redesigned the way the wing attached to the plane.

February 9th flight

For this flight, we had our new method for attaching the wing finished. This attachment was modeled after the method how the skyline champ attaches its wings and still included the rubber bands. We also mounted the IR thermopiles and were able to calibrate them on the plane and test their functionality. For this flight there was a light crosswind making it difficult to take off. The goal of this trip was to test our previous modifications and to get the airframe to fly more stable and

predictable. On the final take off Mr. Gammon had trouble getting it off the ground and it resulted in an aborted sidewise landing breaking the wheel hub. Our major accomplishment during this flight was flying RC mode through the decoder board and autopilot.

February 15th flight

In addition to flying the telemaster we were also about to break in the other two engines for the Skyline Champs. We added the new wheels on the telemaster. Unfortunately the new wheels were not as good as the original and prevented us from taxing straight, making lift offs challenging. We also discovered that one of the wings had become twisted when we re-heat shrunk the covering, which made our plane turn to the right when cruising. We also tested Auto 1 mode during flight. During auto 1 the airplane turned to the right however the twisted wing could have caused that. We also lost our downlink before the flight so no definitive data could be gathered to help tune it.

February 18th flight

For this flight, we straightened out the wheels, and fixed the twist in the wing. We also put our new decoder board in the plane. We also have some crosswind during our take off. After the first flight we discovered that it interpreted the signals differently from the original board and was having over flow issues on the rudder control. Because of this we were not able to test Auto 1 again. After changing it back to purely RC with out routing the signal through the decoder board, we attempted to have a second flight. However it was caught be a cross wind before we fully got up and off the ground. The wind caused the plane to roll to the right to the point the wing caught the ground and caused it to hit the ground sideways ripping off the landing gear and the bottom floor of the airplane. Fortunately the aluminum frame saved most of the plane making the damage minimal.

Appendix G: User Manual

Paparazzi Autopilot User Manual

This User manual is written to guide the user through configuration and operation of both the paparazzi Ground station software and modifying and downloading the configuration files necessary for the autopilot board.

Autopilot

The autopilot board we used for our MQP was a YAPA2. The documentation for the board can be found here. http://paparazzi.enac.fr/wiki/YAPA/v2.0

From the vendor the paparazzi board does not come with an usb bootloader installed on the device. The bootloader provides the computer a ttyUSB0 serial port for downloading to the device. Without the bootloader the device will only be able to be programed through the USART1 port labeled as the 'GPS' pins.

If you plug the autopilot into the computer and then supply power to the pins labeled 'bat (6v-18v)' and then open up a terminal window and path to:

Cd /dev

There should be a ttyUSB0 available. If it is not there and the light on the board is not blinking, you will need to install the usb bootloader. If there is a ttyUSB0 you may skip the boot loader installation guide.

USB Bootloader Instructions

1. Obtain a FTDI 3.3 usb cable (http://www.sparkfun.com/products/9717). This cable is recommended because the FTDI cables have known working drivers for linux operating systems.

2. Plug the FTDI cable like below.
INSERT PICTURES

3. Plug in FTDI cable in to usb port on computer

4. In order to have the processor receive the bootloader code, the boot pin on the processor needs to be connected to ground. Near the xbee headers there is an area labeled 'boot'. Soldering a wire bridging the two vias will work. After bootloader installation make sure you remove the wire.

5. Power on device

This must be done using a current limited power supply with a voltage between 7-10v

6. Open up a linux terminal and type
> cd /dev
and then type
> ls
This will display all of the devices currently available to the computer (including internal)
You should be able to see ttyusb0 as a device. If not unplug then plug in the FTDI usb cable again.

7. Now type

cd /home/<username>/paparazzi
Then
> make upload_bl PROC=GENERIC
This will upload the usb boot loader to the device, on success the terminal should display

Synchronizing. OK
Read bootcode version: 2.12.0
Read part ID: LPC2148, 512 kiB ROM / 40 kiB SRAM (67305253)
Sector 0: ....
Sector 1: ....
Download Finished... taking 6 seconds
Now launching the brand new code
ioctl get failed
ioctl set ok, status = 0
ioctl get ok, status = 2
ioctl get ok, status = 2
ioctl set ok, status = 2
ioctl set ok, status = 0
ioctl set ok, status = 0

Now you should be able to plug in the usb into mini usb connector on the paparazzi board, power on the device and see in the dev 'ttyusb0'.

Autopilot Connection

Now that the autopilot board has an usb bootloader, the ground station software has the ability to download new code to it.
To upload new code:

1. Connect the usb to the mini usb on the autopilot and make sure power is off.
2. Plug in the usb to the computer.
3. Turn on power to the autopilot.
4. No lights should blink on the autopilot, which signifies that the autopilot is ready to receive new code.
5. Open up the paparazzi ground station.
6. Select your airframe and set your target to 'ap'.
7. Run Clean and then Make. Make will take in all of the configuration files and compile the c code to download to the autopilot.
8. Select Upload to upload to the autopilot. If it does not work make sure the autopilot is powered

To connect to the autopilot via USB to receive Telemetry:

1. Within the airframe configuration file, the following must be present within the

   firmware section :
   <subsystem name="telemetry" type="transparent_usb"/>
   </subsystem>

2. .Build and Upload the new configuration file to the autopilot
3. Turn on the autopilot before plugging in the usb so that the autopilot does not go into the bootloader sector
4. Connect to the autopilot using "Flight USB-serial@9600 . For this configuration the autopilot will show up as a ttyACM0 device under /dev/.

Xbee Connection:

To connect to the autopilot via xbee radios the radios must be loaded with firmware first.

Acquiring X-CTU

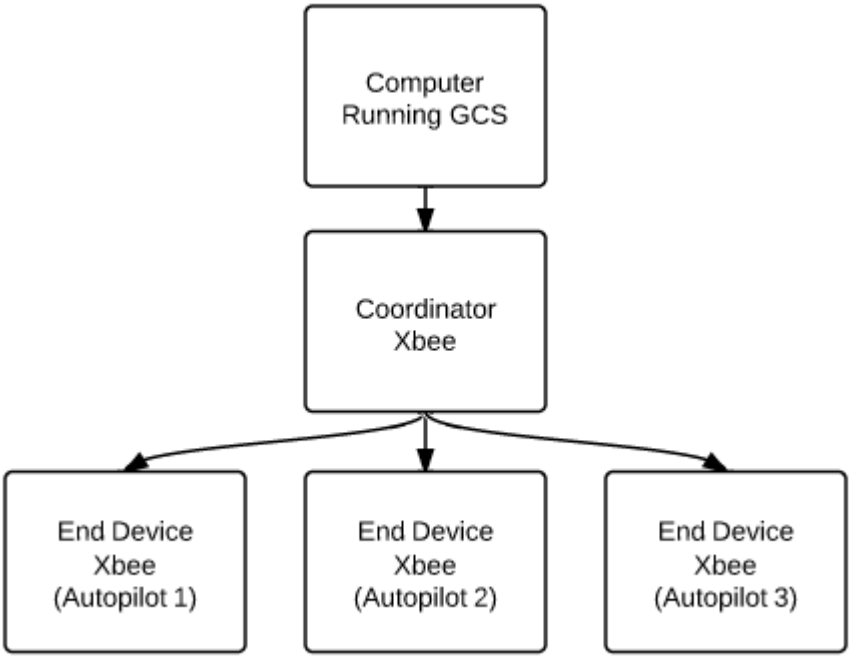
X-CTU is a windows software tool provided by Digi that allows you to update the firmware to the Xbee radios and then configure them as necessary. X-CTU can be obtained from the website http://ftp1.digi.com/support/utilities/40002637_c.exe and then executed with Wine on Linux.

Since X-CTU is windows based it expects the Xbee device to be connect to a COM port. To get around this in Linux you can make Wine see /dev/ttyUSB0 as a COM device by using the following commands.

>cd ~/.wine/dosdevices
>ln -s /dev/ttyUSB0 ~/.wine/dosdevice/com10
Now when you connect the Xbee to the computer Wine will set up COM10 as the xbee.

Using X—CTU:

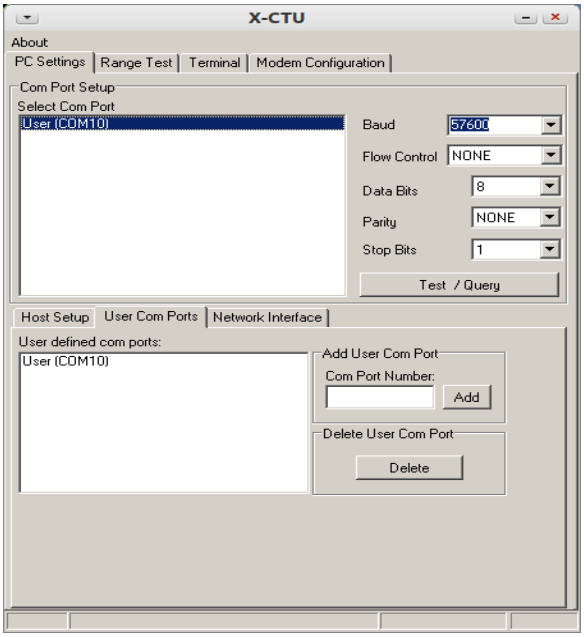
With X-CTU set up to run on Linux the next step is to update the firmware on the Xbee radios. All Xbee radios that are in use must be updated. Below is a basic diagram of the different Xbee radios used with Paparazzi.



Updating the firmware:

Note: Keep track of which xbee radio is the coordinator and which are the end devices. If they are not marked they are easy to lose track of them when updating.

1. Place the Coordinator Xbee radio in the usb connector and then plug in the usb into the computer
2. Execute X-CTU with Wine
3. Select the user comports tab and then add comport 10



4. Make the comport active by clicking it within Select Comport Window and then set the baud rate to 57600 (Note if the Xbee radio is new the default baud rate may be different)
5. Select the modem configuration tab
6. Hit Read to check the current firmware
7. Confirm the following settings
    - PAN ID : 3332
    - Destination Address High : 0
    - Destination Address Low : FFFF
    - Coordinator : 1 (Coordinator)
    - Interface Data Rate : 57600
    - API Enabled : 1- API Enabled
8. Now hit Write to update the Xbee
9. Place an End device into the usb connector
10. Hit Read to update the window with the end devices configuration
11. Confirm the following Settings

    - PAN ID : 3332
    - Destination Address High : 0
    - Destination Address Low : FFFF
    - Coordinator : 0 (End Device)
    - Interface Data Rate : 57600
    - API Enabled : 1- API Enabled

The only difference is the destination that this is an end device. Having the Destination address set to FFFF has the xbee radios transmit to all radios in the area. Since this network is in a star topology with only end devices talking to the coordinator other end devices will ignore each other. The GCS will receive packets from the Autopilots and parse them based on the aircraft identification number for each autopilot.

Now that the radios are loaded with the correct firmware the following change is required within the airframe configuration file to make use of the Xbee API.

```
<subsystem name="telemetry"     type="xbee_api">
        <configure name="MODEM_BAUD"        value="B57600"/>
 </subsystem>
```

Build and upload the new configuration and connect to the autopilot using "Flight USB-XBee-API@57600"

If "Flight USB-XBee-API@57600" is not an option it can be added to your control panel with the following lines:

```
<session name="Flight USB-XBee-API@57600">
<program name="Data Link">
        <arg flag="-d" constant="/dev/ttyUSB0"/>
```

```
        <arg flag="-transport" constant="xbee"/>
        <arg flag="-s" constant="57600"/>
</program>
<program name="Server"/>
<program name="GCS"/>
<program name="Messages"/>
</session>
```

To use multiple planes connect to the first plane then go back and select your second plane and then connect using the same method.

Ground control station

The ground control station (GCS) provides the visual feedback of the UAV.A detailed explanation of it can be found at the paparazzi wiki located here:

http://paparazzi.enac.fr/wiki/GCS

A key feature to note is the settings tab. This allows the user to update of values defined in the airframe configuration file during runtime. This is important for modifying the control characteristics of the UAV and any neutrals of the sensors during flight.

The flight plan loaded is dependent on the flight plan configuration file. The home way point can be defined at the top of the configuration file and when you load the GCS the home way point will be located there. By going to the tool bar at the top of the GCS and then selecting maps, you can select the source of the map (default is google) and then load the tile corresponding to your waypoints.

Configuring Plane Airframe Selection:

To add a new airplane to the selectable list:

1. Open /paparazzi/conf/conf.xml
2. Add an additional aircraft to the list of aircraft changing the following
   - Name : New name for your aircraft
   - Ac_ID : A new id number for your plane, must be different from the others
   - Airframe : default airframe file
   - Radio : default radio configuration file
   - Telemetry : default telemetry configuration file
   - Flight_Plan : default flight plan file
   - Settings : default settings file

Configuring Connection and GCS:

To add or modify a new connection setting:

1. Open /paparazzi/conf/control_panel.xml
2. Find the “sessions” section
3. Each session contains different ground station programs, the required ones are:

- Data Link: The data link connects to the autopilot Typical flags used are :

<arg flag="-d" constant="/dev/ttyUSB0"/> (location of the device to be connected to)
<arg flag="-transport" constant="xbee"/>
(type of protocol for communications)
<arg flag="-s" constant="57600"/> (baudrate)

- Server: Used to continuously record the messages received from the autopilot. They are stored in /paparazzi/var/
- GCS: The ground control station. Provides a user interface to the autopilot system
- Messages: Provides a user interface to reading incoming messages from the autopilot. Useful for debugging.

Multi-plane Simulation:

To perform a multi-plane simulation:

1. Select the first plane by using the airframe selection of the left.
2. Select build target to be “Sim”
3. Run make
4. Then select simulation under execution then hit connect.
5. The GCS and the mni Simulator window will now pop up, simply mimize them and return to the paparazzi launcher and select the second airframe.
6. Make and then execute in the same fashion
7. Now a second GCS window and simulator will pop up except this GCS window will contain both planes that were executed.

The GCS should look similar to the image below.

Radio Controller Interface

Paparazzi autopilots provide a radio controller interface system for manual override of the UAV during flight. This is a critical safety feature for the autopilot and also required by FAA regulations. The autopilot expects a pulse position modulation (PPM) from a radio controller’s receiver. The radio configuration file (located in /paparazzi/conf/radios/) defines each of the channels and the time ranges for each.

Some systems currently use a PPM communication system and it’s as simple as wiring the receiver’s PPM line to the R/C PPMIN pin on the autopilot board. Unfortunately not all systems utilize PPM or the PPM trace is difficult to access in the receiver.

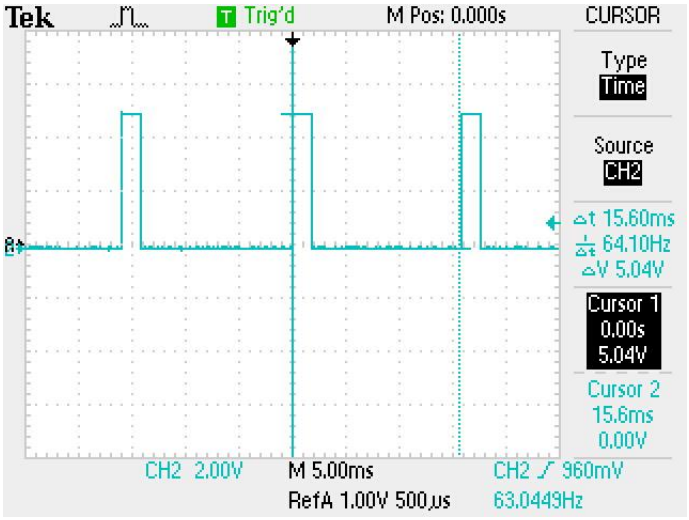
The MeekPE encoder, an open source PPM encoder board, provides a solution by taking the servo channels and mixing them into a PPM signal. This allows for a PPM to be generated from any style of receiver system and without modification of the receiver.
MeekPE boards are currently available at batchpcb at the following link http://batchpcb.com/index.php/Products/43035, and the bill of materials is listed at the following site http://paparazzi.enac.fr/wiki/MeekPE_PPM_Encoder_Board.

Measuring the Radio Receiver and PPM Signal:

1. Bind the radio controller to the radio's receiver (instructions depend on the system) and make sure all of the trims on the radio are zero.
2. The servo commands are not transmitted at the same time by the radio and come in sequentially. By connecting two oscilloscope probes to two of the servo channels on the receiver you should be able to tell that one comes after the other.

   Use as many oscilloscope probes available to determine the last channel transmitted to the receiver.

   The last servo channel is what is connected to the last servo connector on the MeekPE board.
3. Connect the oscilloscope probe one of the servo channels (make sure you ground the oscilloscope probe to the receiver).
4. Measure the neutral servo value for the corresponding control surface



Neutral PWM Measurement

5. Measure the minimum and maximum for the corresponding control surface.
6. Notate the values along with the servo channel. These values will be used by the autopilot configuration file to define the ranges of the servos.
7. Repeat 3-6 for all of the servo channels.

8. Connect the receiver channels to the MeekPE board. The order and amount of channels does not matter except the last servo channel from the last step must be plugged into the last servo of the MeekPE.
9. Hook up an oscilloscope probe to the output PPM of the MeekPE and make sure the MeekPE board is powered, either from the receiver or through the 5v line on the MeekPE board.

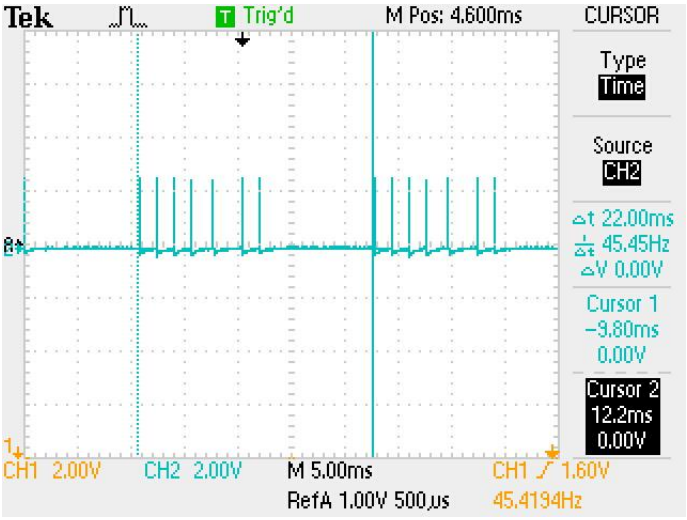


Figure 103:PPM Signal

10. By moving the sticks on the radio controller you should see the packet change widths on the oscilloscope. Find the channel that is first on the left and notate which control surface it affects (aile/elev etc).
11. Measure the width of the channel when the stick is in the neutral position.

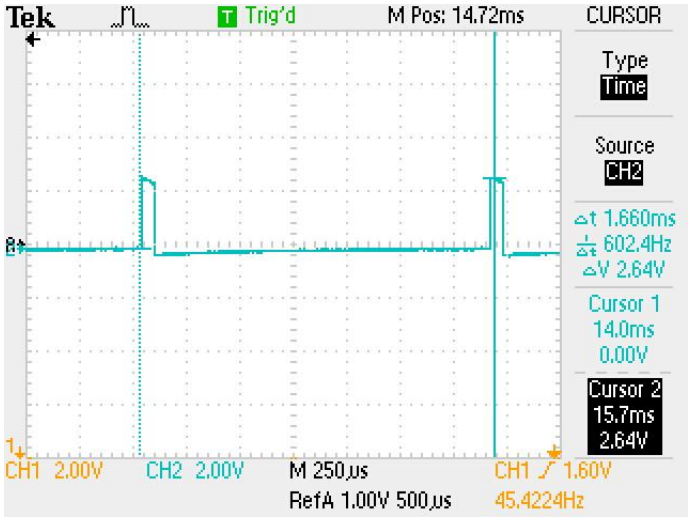


Figure 104: Neutral Channel One

12. Measure the width of the channel when the stick is in one extreme and then the other extreme.

13. Take note of these values as they will have to be entered into the radio configuration file.
14. Now measure the rest of the channels. Notating all of the values.
15. Before disconnecting we need to take the following measurements.
    1. Synchronization Packet
    2. Max packet Size

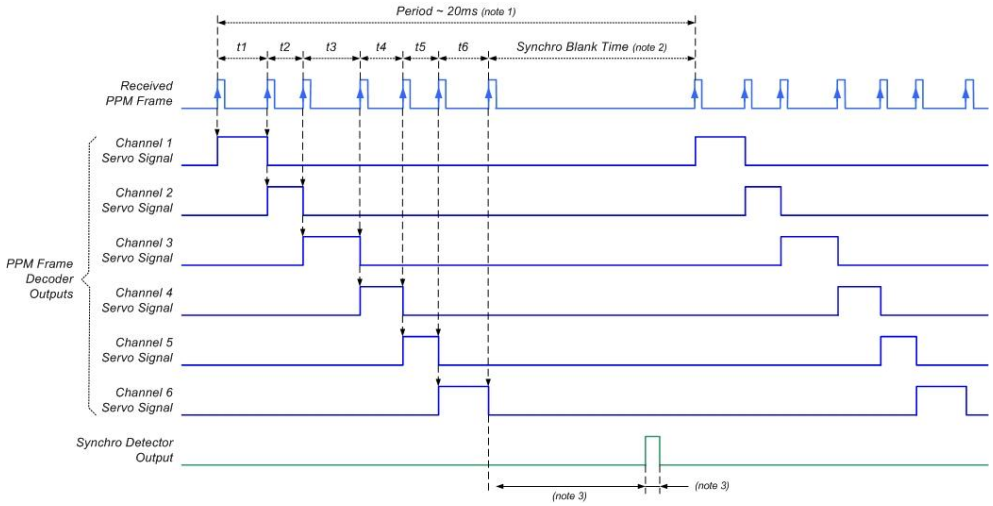


Figure 105: PPM Structure

16. Now open the configuration file associated with your radio system. If one does not exist, copy an existing one and make the same following changes.
17. Add in the Channel readings to your radio configuration file. For example if the first channel is AIL and the readings were 1220, 1620, and 2020 for min, neutral, and max it will be set as the following:

    <channel ctl="AIL" function="ROLL" min="1220" neutral="1620" max="2020" average="0"/>
18. Below is an example of a completed radio configuration file:

```
<!DOCTYPE radio SYSTEM "radio.dtd">
<radio name="AR6200" data_min="900" data_max="3100" sync_min ="5000" sync_max
="15000" pulse_type="POSITIVE">
 <channel ctl="AIL" function="ROLL"           min="1220" neutral="1620" max="2020"
average="0"/>
 <channel ctl="AUX" function="FLAP"       min="1220" neutral="1620" max="2020"
average="0"/>
 <channel ctl="ELE" function="PITCH"      min="1220" neutral="1640" max="2040"
average="0"/>
 <channel ctl="GEAR" function="MODE"      min="1100" neutral="1760" max="2060"
average="0"/>
 <channel ctl="RUD" function="YAW"           min="2260" neutral="2660" max="3080"
average="0"/>
```

```
 <channel ctl="THRO" function="THROTTLE"      min="1100" neutral="1520" max="2020"
average="0"/>
</radio>
```

The order here matters. It is the order that the channels are within the PPM. If this is setup wrong some control surfaces may not work or will be swapped.

Configuring Control Surfaces:

Now that the radio configuration file is set up, the next step is to define the control surfaces within the airframe configuration file.

Servo Definition:

- First you define the servos and their names.

```
<servos>
<!--define the servo ports and their min/neutral/max values-->
<servo name="MOTOR" no="0" min="1000" neutral="1440" max="1960"/>
<servo name="AILEVON_LEFT" no="1" min="1000" neutral="1500" max="2000"/>
<servo name="RUDDER" no="2" min="1040" neutral="1440" max="1840"/>
<servo name="AILEVON_RIGHT" no="3" min="1000" neutral="1500" max="2000"/>
<servo name="ELEVATOR" no="4" min="1880" neutral="1440" max="1040"/>
</servos>
```

- The servo name is the "object" name for that servo referenced later.
- The no (number) sets which servo output the channel will be outputted on.
- The min /neutral / max values are the numbers measured from the output of the receiver.
- To flip the direction of the servo, swap the value for min and max.

Command Law Definition:

The paparazzi autopilot interfaces through the control surfaces using the command laws. The command laws used by the autopilot are ROLL, PITCH, YAW, and THROTTLE. The command laws need to be set to the servos we previously defined. Below is an example:

```
 <command_laws>
   <!--Connect the servos to the corresponding command laws of the autopilot-->
   <set servo="MOTOR"        value="@THROTTLE"/>
   <set servo="AILEVON_LEFT"  value="@ROLL"/>
   <set servo="AILEVON_RIGHT" value="@ROLL"/>
   <set servo="ELEVATOR" value="@PITCH"/>
   <set servo="RUDDER" value="@YAW"/>
 </command_laws>
```

Here is where mixing can be done similar to options presented through the radio transmitter system.

```
 <command_laws>
   <let var="aileron" value="@ROLL * AILEVON_AILERON_RATE"/>
   <let var="elevator" value="@PITCH * AILEVON_ELEVATOR_RATE"/>
   <set servo="MOTOR" value="@THROTTLE"/>
```

```
  <set servo="AILEVON_LEFT" value="$elevator + $aileron"/>
  <set servo="AILEVON_RIGHT" value="$elevator - $aileron"/>
 </command_laws>
```

This allows for mixing the elevator with the ailerons which allows for more stable turns.

RC Command Definition:

During AUTO1 the pilot has control over the ROLL and PITCH commands (ailerons and elevator). To add throttle and rudder add the following section:

```
 <auto_rc_commands>
  <!--Definitions of the RC commands during AUTO1-->
  <set command="THROTTLE"  value="@THROTTLE"/>
  <set command="YAW"       value="@YAW"/>
 </auto_rc_commands>
```

To test throttle during AUTO1 the command has to be removed from this section so that the autopilot has full control over the throttle.

To confirm that the command laws were defined properly set up the plane with the autopilot and all of the control surfaces plugged in their corresponding channels. When the autopilot is turned on and the radio switched into MANUAL mode all of the control surfaces should react to the radio controller. When switched into AUTO1 the plane should start attempting to stabilize itself.

Note: If indoors and using the IR Thermopile system the control surfaces may act erratically. Make sure the vertical sensor sees ground using your hand.

In AUTO1 the user should still have control over the ailerons, elevator and then any of the surfaces defined in the auto_rc_commands section.

In AUTO2 the user should not have any control over the control surfaces.

Three Switch Position Setup:

A three position switch needs to be dedicated to change the mode of the autopilot (Manual / AUTO1 / AUTO2).

We used a Spektrum dx6i which did not provide any three position switches. An alternative solution was to mix the gear switch with the AUX1 switch so that one switch selected autopilot or manual mode, and then the AUX1 switch selected AUTO1 or AUTO2.

This channel was then mixed to the GEAR channel which was defined as MODE within the radio configuration file.

Debugging Radio Communications:

To determine if the autopilot board is properly receiving all radio channels from the radio receiver:

1. Activate the PPM messages within the Telemetry messages (see Telemetry Messages for definition). The default telemetry message packet does not contain it. To add it in go to your airframe configuration file and add the following line under the “fixedwing” section.

<define name="TELEMETRY_MODE_FBW" value="1"/>

2. After communicating to the board you should see PPM messages within the messages window (See GCS configuration if not)
3. This window shows the PPM packet parsed into the corresponding channels.
4. There should be changing values in the 20000 range for each of the channels you defined. If there are unchanging channels or there are no channels check your connection and your channel measurements.
5. The GCS contains a RC status light, if it is Green you know it is all set, else wise it is red and unable to properly read RC messages.

Developers Guide

Now that you have paparazzi up and running, the goal of the developers section is to provide insight on where major components of the source code are.

Configuration Files:

As seen so far, the major way modify the code you download to the autopilot is driven by the configuration file system. All of the configuration files are stored in /paparazzi/conf

Airframe:

The airframe configuration file consists of the assignment of configuration files for all subsystems and sensors. Also included are the definitions of the control surfaces and their behaviors, and the defaults for many flight parameters. A configuration file system allows Paparazzi to be flexible with new sensors or other subsystems, by allowing the user to define new configuration files and associating them with their corresponding sensors or subsystems within the airframe file.

Telemetry:

The telemetry configuration file defines the messages used for plane to base station communications. All of the messages within the telemetry are transmitted by the UAV and then displayed at the ground station within the messages window. This allows for debugging of the plane and its various subsystems on the ground and during flight. All messages are saved for review after the flight. Each type of message has a period defined. This allows for less important messages to be slowed down reducing data transmission.

```
<message name="AIRSPEED"          period="1"/>
<message name="ALIVE"             period="5"/>
<message name="GPS"               period="0.25"/>
<message name="NAVIGATION"        period="1."/>
<message name="ATTITUDE"          period="0.5"/>
<message name="ESTIMATOR"         period="0.5"/>
<message name="ENERGY"            period="2.5"/>
<message name="WP_MOVED"          period="0.5"/>
<message name="CIRCLE"            period="1.05"/>
```

<message name="DESIRED" period="1.05"/>
<message name="BAT" period="1.1"/>

Above is an example of some of the messages that are sent from the default telemetry file. After testing the autopilot in the air we will begin removing unnecessary messages.

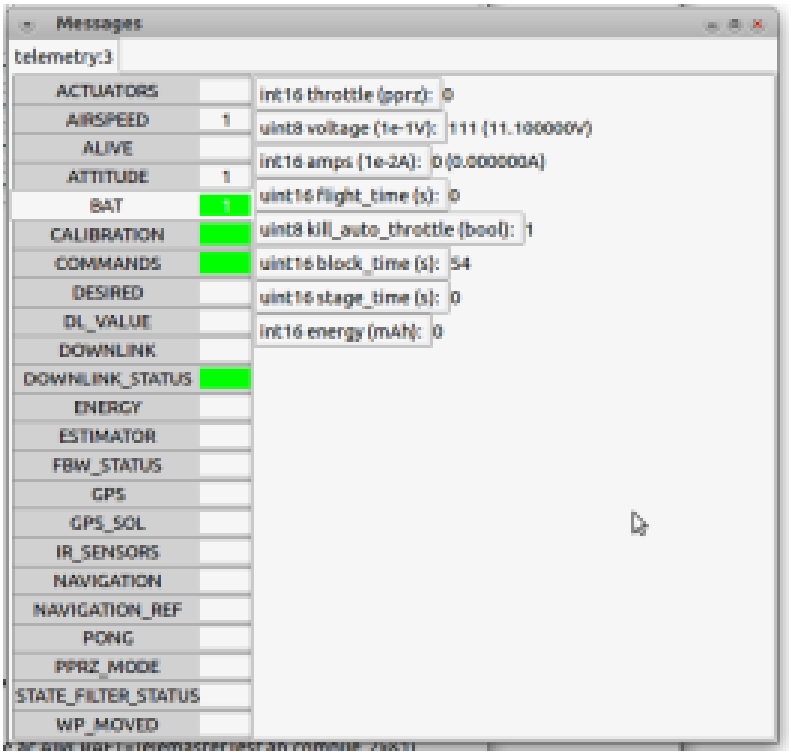


Figure : Above is an example of the messages window provided by the ground station to view incoming messages from the UAV. The “bat” message is currently selected and shows values such as voltage and current from the battery.

```
93.355 8 NAVIGATION 2 1 -2.4 4. 0. 21.9 0 0
93.375 8 GPS 3 26853456 468411392 2955 160370 37 -28 1662 252373750 19 1
93.423 8 WP_MOVED 15 268414.5 4684145.5 234.559998 19
93.424 8 CALIBRATION 0. 0
93.519 8 AIRSPEED 15. 14.5 14.5 6.
93.603 8 NAVIGATION 2 1 -2.4 4. 0. 21.9 0 0
93.631 8 GPS 3 26853456 468411392 2988 160290 34 -32 1662 252374000 19 1
93.632 8 NAVIGATION 2 1 -2.4 4. 0. 21.9 0 0
93.719 8 ATTITUDE 1.925012 5.238955 -1.318443
93.823 8 ESTIMATOR 0. 0.
93.831 8 DESIRED 0. 0.24 0. 0. 0. 45. 0.
93.855 8 NAVIGATION 2 1 -2.4 4. 0. 21.9 0 0
93.875 8 GPS 3 26853456 468411392 3001 160250 32 -16 1662 252374250 19 1
93.919 8 WP_MOVED 0 268579. 4684152. 234.559998 19
94.043 8 DOWNLINK_STATUS 94 32917 1471 0 469. 20. 17.28
94.103 8 FBW_STATUS 2 0 1 126 0
94.111 8 NAVIGATION 2 1 -3. 4. 0. 25. 0 0
94.131 8 GPS 3 26853400 468411392 3153 160160 31 -36 1662 252374500 19 1
```

Figure : Above is a sample data file collected during tested. The messages above were selected by the telemetry file and then recorded when received from the UAV.

Radio:

The radio configuration file defines how the PPM message from a radio receiver is parsed into the autopilot system. This setup process is further described under “Measuring the Radio Receiver and PPM Signal:”

Settings:

The settings configuration file allowed us to change values during flight time for numerous subsystems. This expedited the testing process as it allowed us to change flight parameters from the ground as opposed to landing the UAV, and uploading new changes between each flight. Values defined in the settings file are shown in the settings tab within the ground station.

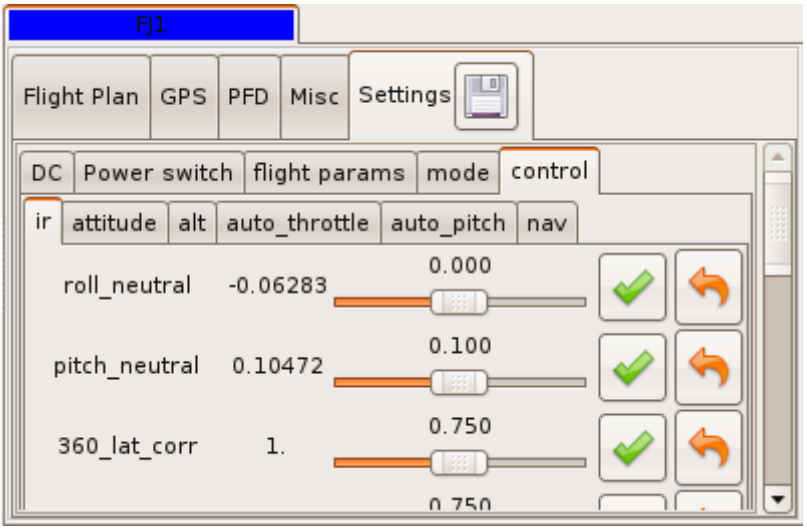
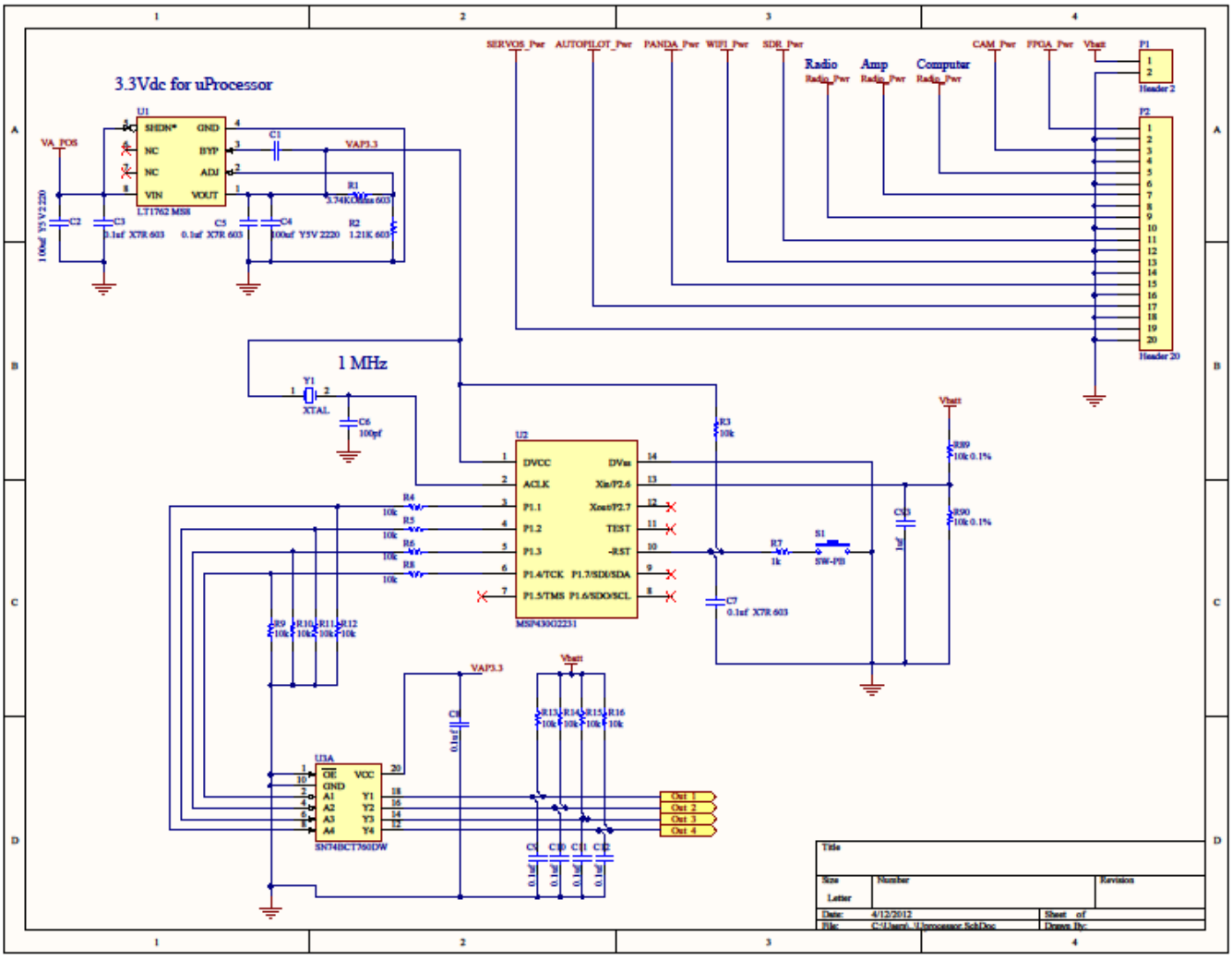


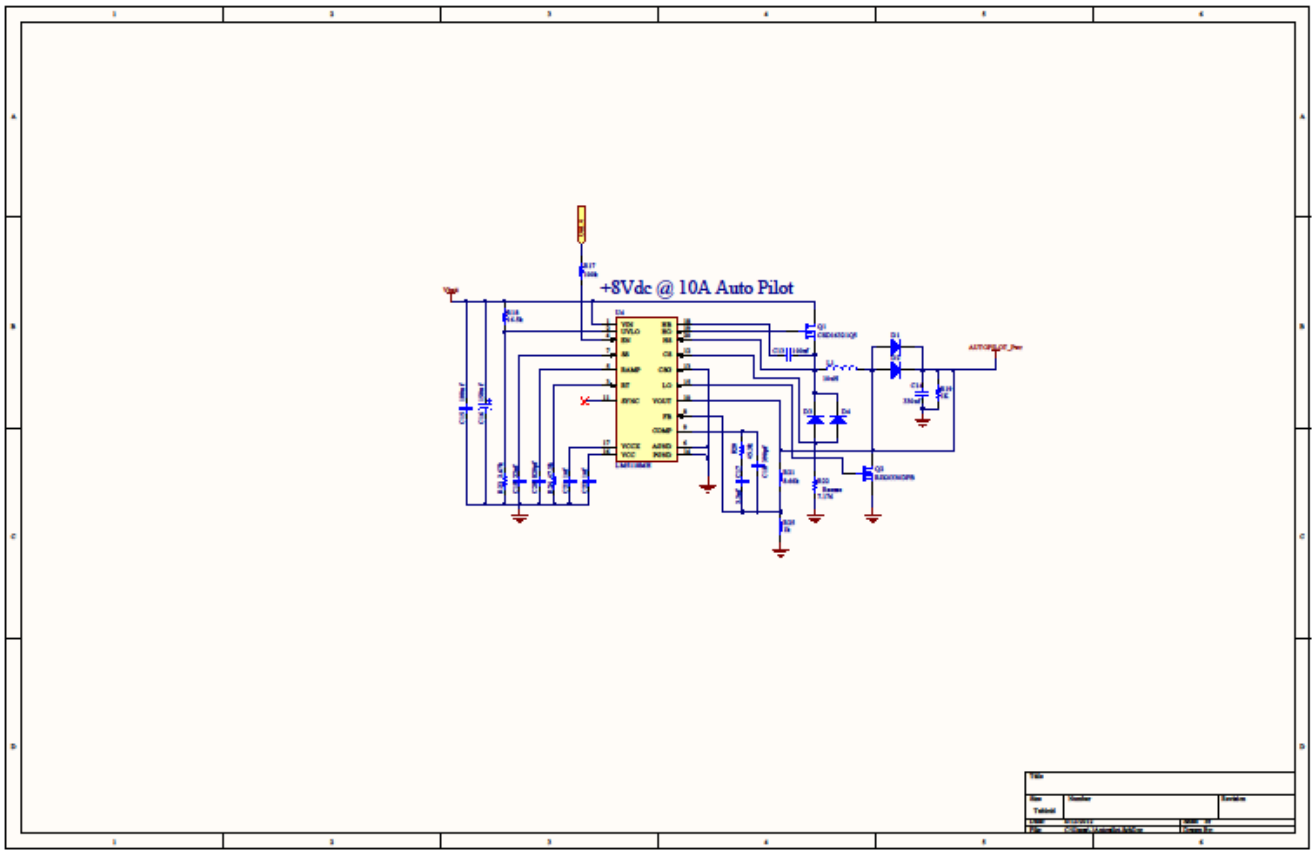
Figure : Settings Tab within Ground Station

The configuration files only set values and define flags for the autopilot code which is all stored within /paparazzi/sw/airborne

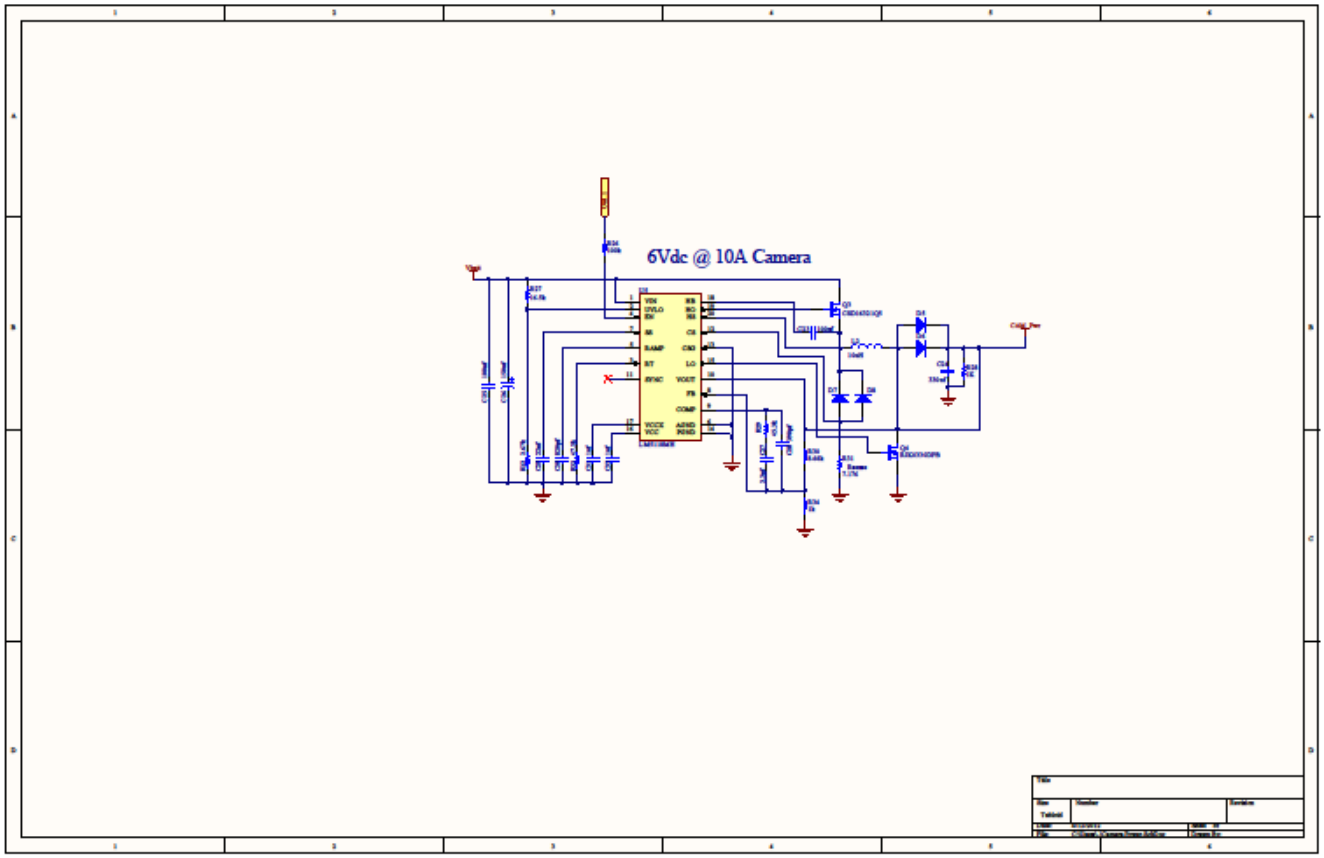
Appendix H: Power board schematic



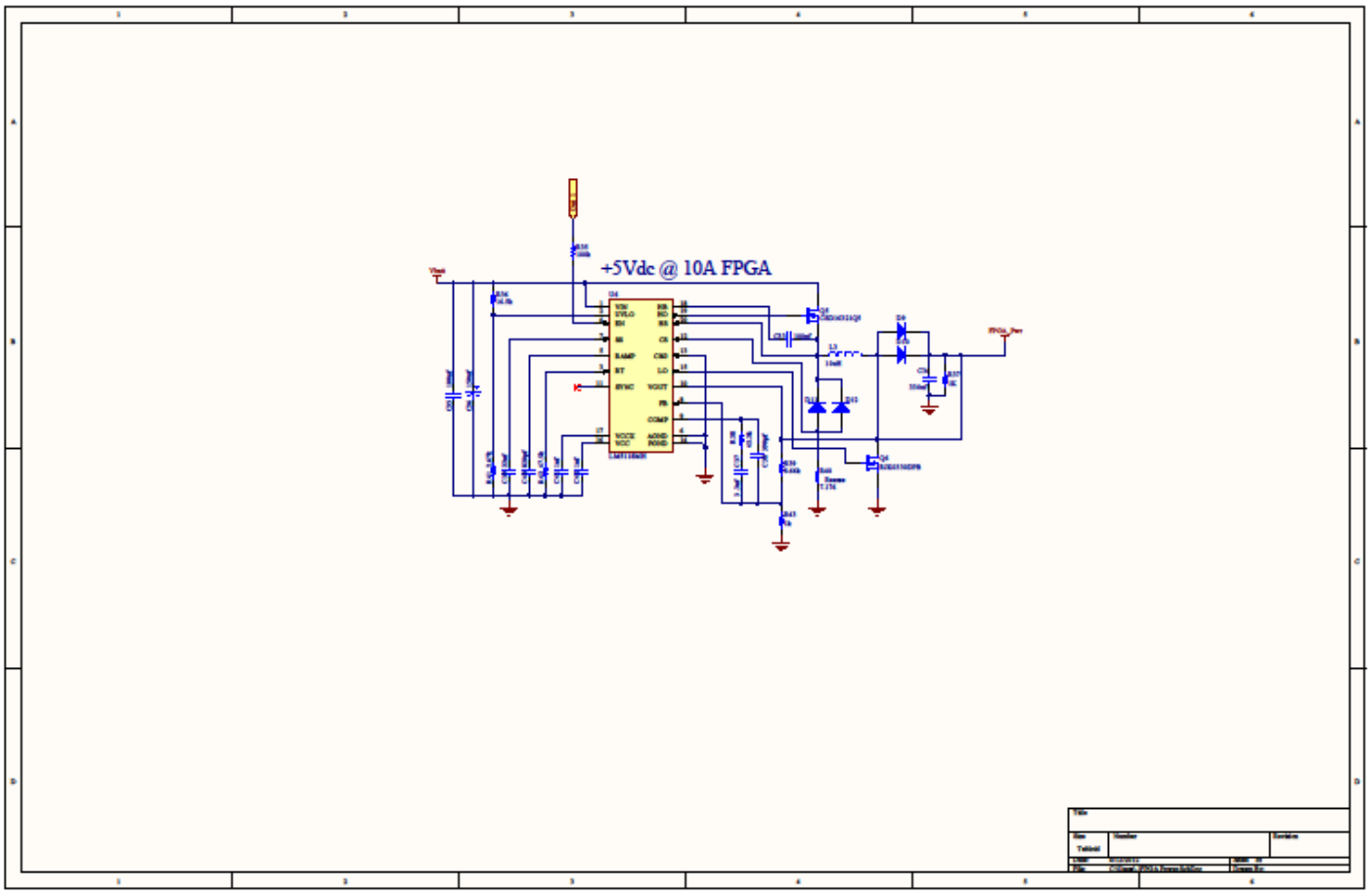
Schematic 1: Power board control logic



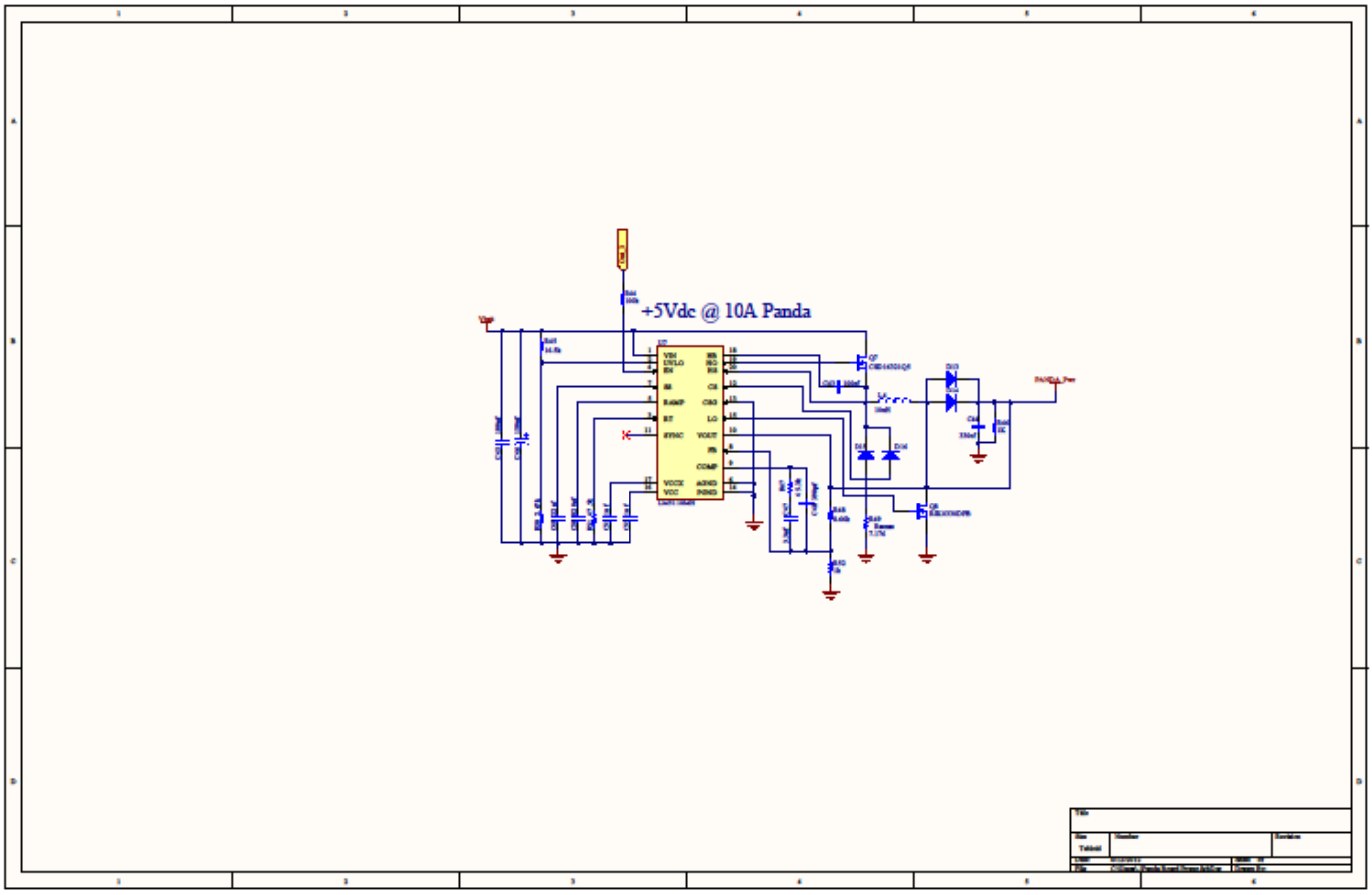
Schematic 2: Regulator circuit for autopilot



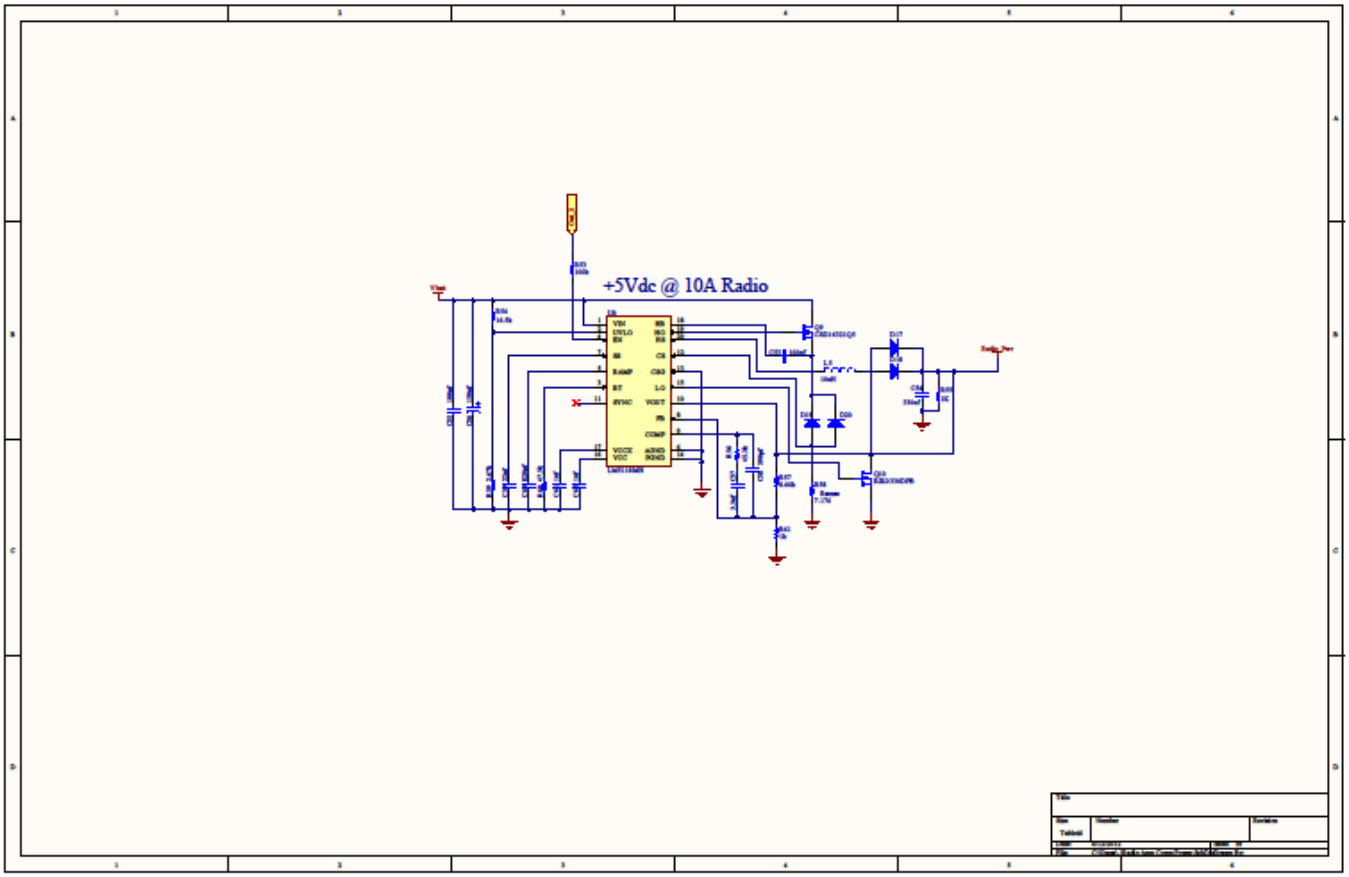
Schematic 3: Regulator circuit for camera



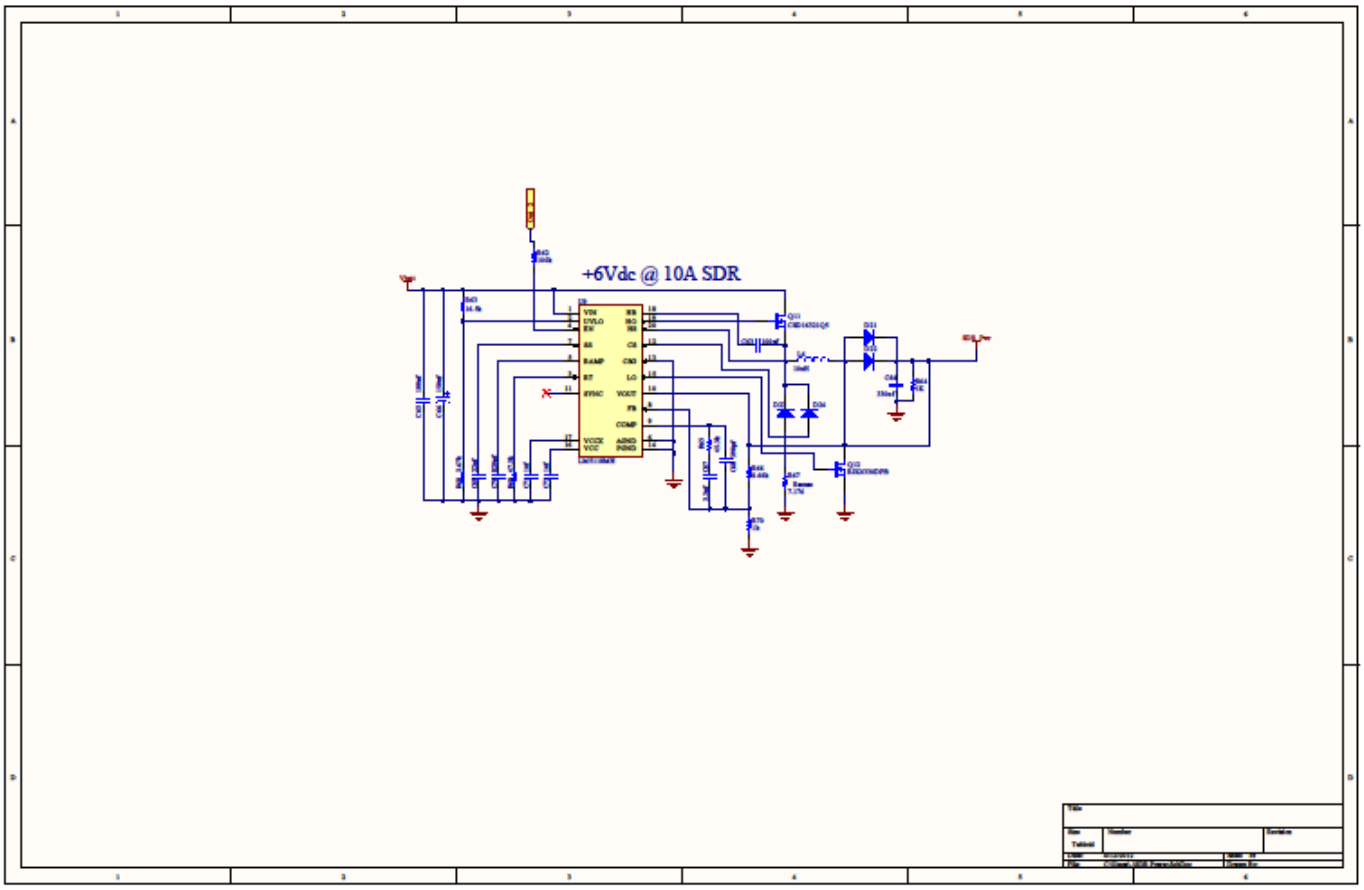
Schematic 4: regulator circuit for FPGA



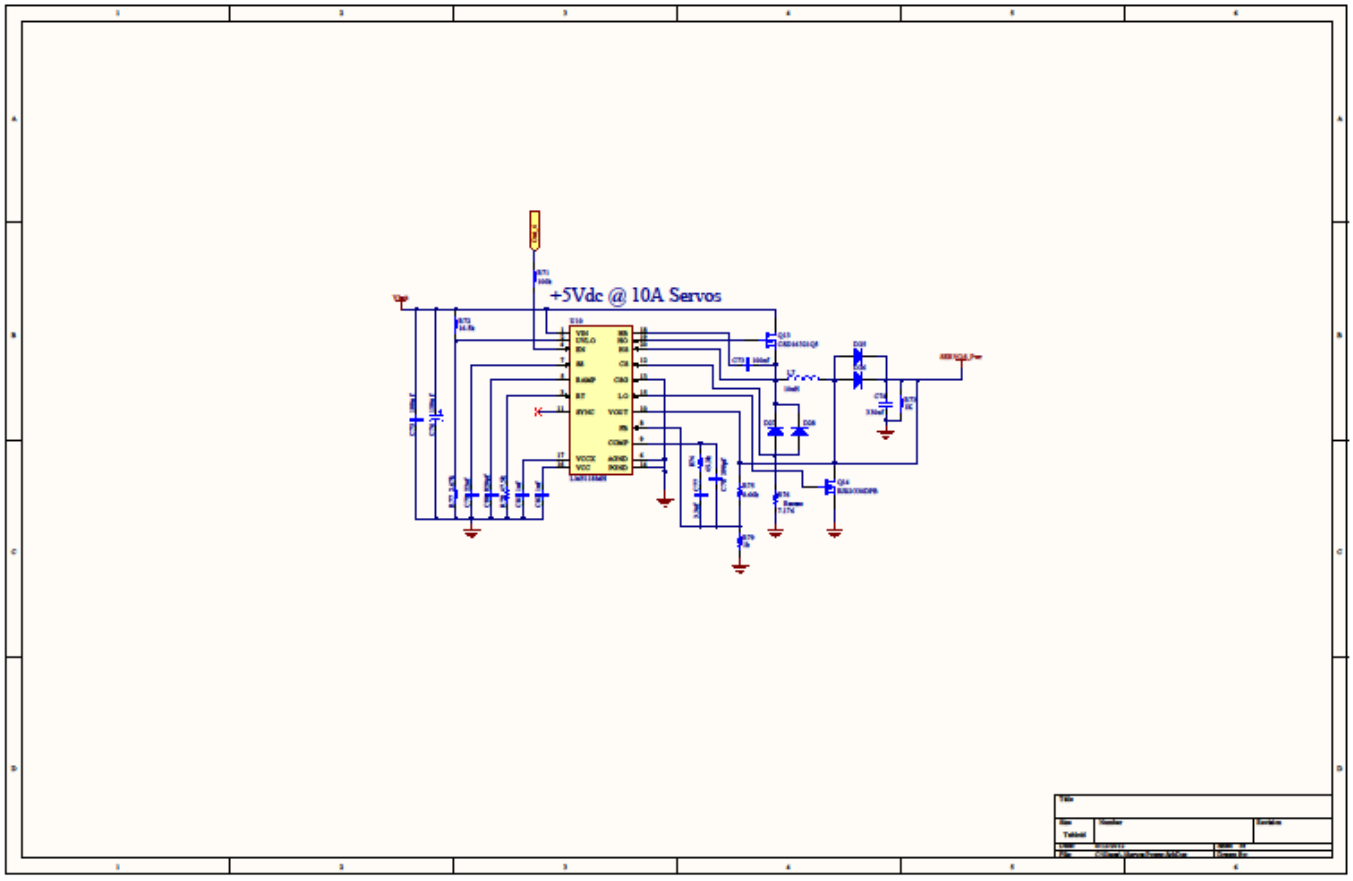
Schematic 5: regulator circuit for Panda board



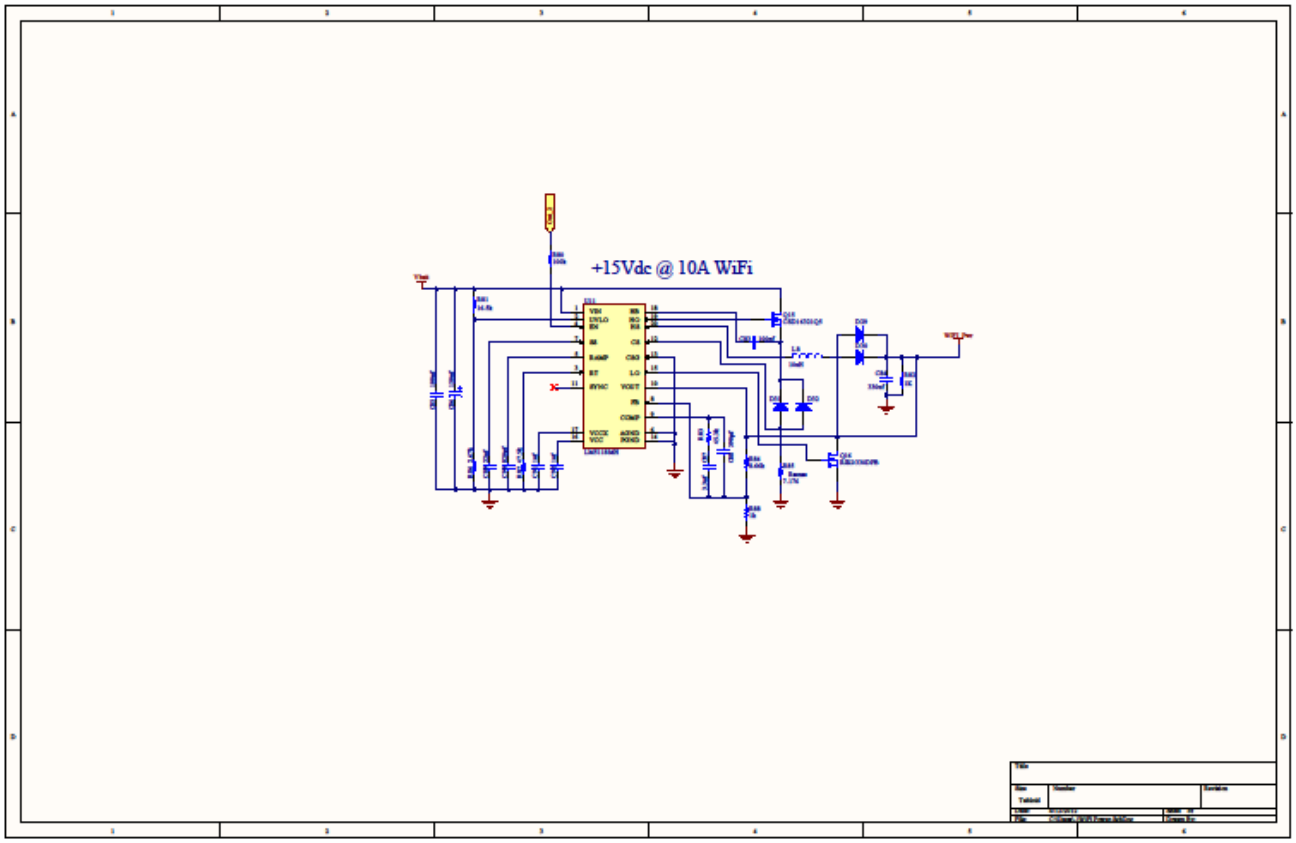
Schematic 6: Regulator for radio support hardware



Schematic 7: regulator for the USRP2 SDR radio



Schematic 8: Independent regulator for servos



Schematic 9: Regulator for the WiFi modules

Appendix I: LM5118 Data Sheet

National Semiconductor

April 30, 2008

LM5118
Wide Voltage Range Buck-Boost Controller

General Description

The LM5118 wide voltage range Buck-Boost switching regulator controller features all of the functions necessary to implement a high performance, cost efficient Buck-Boost regulator using a minimum of external components. The Buck-Boost topology maintains output voltage regulation when the input voltage is either less than or greater than the output voltage making it especially suitable for automotive applications. The LM5118 operates as a buck regulator while the input voltage is sufficiently greater than the regulated output voltage and gradually transitions to the buck-boost mode as the input voltage approaches the output. This dual mode approach maintains regulation over a wide range of input voltages with optimal conversion efficiency in the buck mode and a glitch-free output during mode transitions. This easy to use controller includes drivers for the high side buck MOSFET and the low side boost MOSFET. The regulator's control method is based upon current mode control utilizing an emulated current ramp. Emulated current mode control reduces noise sensitivity of the pulse-width modulation circuit, allowing reliable control of the very small duty cycles necessary in high input voltage applications. Additional protection features include current limit, thermal shutdown and an enable input. The device is available in a power enhanced TSSOP-20 package featuring an exposed die attach pad to aid thermal dissipation.

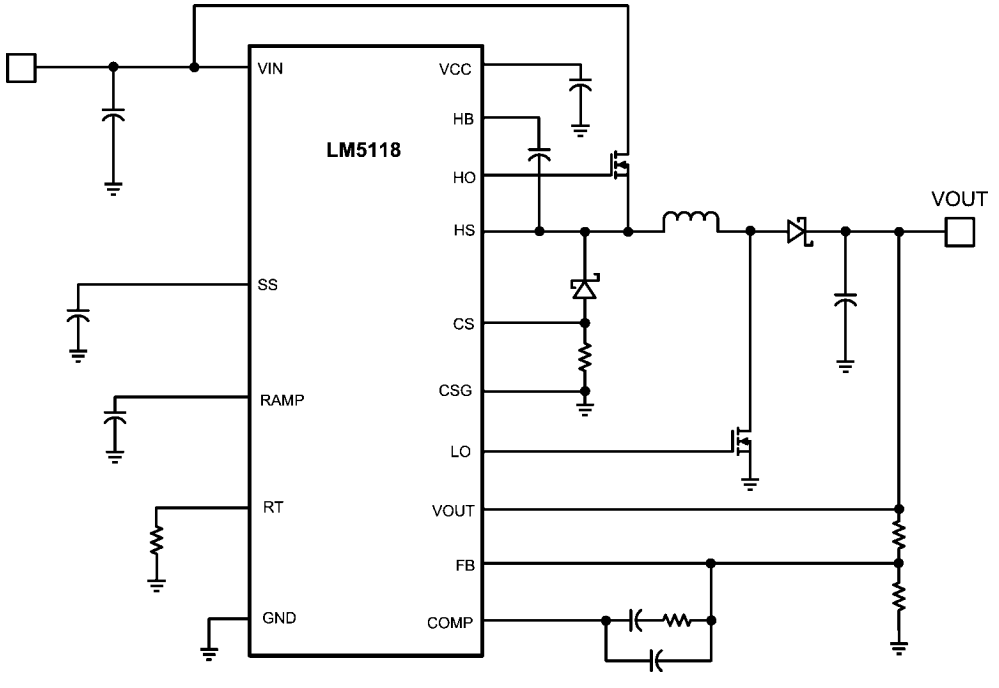
Features

- Ultra-wide input voltage range from 3V to 75V
- Emulated peak current mode control
- Smooth transition between step-down and step- up modes
- Switching frequency programmable to 500KHz
- Oscillator synchronization capability
- Internal high voltage bias regulator
- Integrated high and low-side gate drivers
- Programmable soft-start time
- Ultra low shutdown current
- Enable input wide bandwidth error amplifier
- 1.5% feedback reference accuracy
- Thermal shutdown

Package

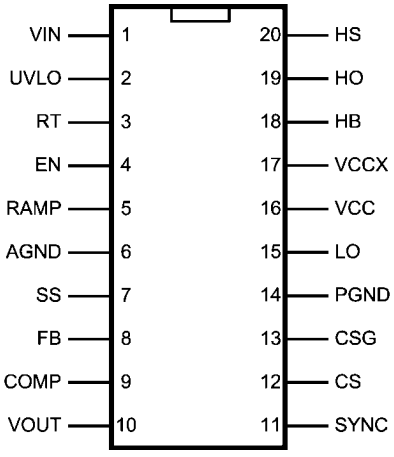
TSSOP-20EP (Exposed pad)

Typical Application Circuit



© 2008 National Semiconductor Corporation

Connection Diagram



Top View
See NS Package Numbers MXA20A

Ordering Information

Ordering Number	Package Type	NSC Package Drawing	Supplied As
LM5118MH	TSSOP-20EP	MXA20A	73 Units Per Anti-Static Tube
LM5118MHX	TSSOP-20EP	MXA20A	2500 units shipped as Tape & Reel

Pin Descriptions

Pin	Name	Description
1	VIN	Input supply voltage.
2	UVLO	If the UVLO pin is below 1.23V, the regulator will be in standby mode (VCC regulator running, switching regulator disabled). When the UVLO pin exceeds 1.23V, the regulator enters the normal operating mode. An external voltage divider can be used to set an under-voltage shutdown threshold. A fixed 5 µA current is sourced out of the EN pin. If a current limit condition exists for 256 consecutive switching cycles, an internal switch pulls the UVLO pin to ground and then releases.
3	RT	The internal oscillator frequency is set with a single resistor between this pin and the AGND pin. The recommended frequency range is 50 kHz to 500 kHz.
4	EN	If the EN pin is below 0.5V, the regulator will be in a low power state drawing less than 10 µA from VIN. EN must be raised above 3V for normal operation.
5	RAMP	Ramp control signal. An external capacitor connected between this pin and the AGND pin sets the ramp slope used for emulated current mode control.
6	AGND	Analog ground.
7	SS	Soft-Start. An external capacitor and an internal 10 µA current source set the rise time of the error amp reference. The SS pin is held low when VCC is less than the VCC under-voltage threshold (< 3.7V), when the UVLO pin is low (< 1.23V), when EN is low (< 0.5V) or when thermal shutdown is active.
8	FB	Feedback signal from the regulated output. Connect to the inverting input of the internal error amplifier.
9	COMP	Output of the internal error amplifier. The loop compensation network should be connected between COMP and the FB pin.
10	VOUT	Output voltage monitor for emulated current mode control. Connect this pin directly to the regulated output.
11	SYNC	Sync input for switching regulator synchronization to an external clock.
12	CS	Current sense input. Connect to the diode side of the current sense resistor.
13	CSG	Current sense ground input. Connect to the ground side of the current sense resistor.
14	PGND	Power Ground.
15	LO	Boost MOSFET gate drive output. Connect to the gate of the external boost MOSFET.

Pin	Name	Description
16	VCC	Output of the bias regulator. Locally decouple to PGND using a low ESR/ESL capacitor located as close to the controller as possible.
17	VCCX	Optional input for an externally supplied bias supply. If the voltage at the VCCX pin is greater than 3.9V, the internal VCC regulator is disabled and the VCC pin is internally connected to VCCX pin supply. If VCCX is not used, connect to AGND.
18	HB	High side gate driver supply used in bootstrap operation. The bootstrap capacitor supplies current to charge the high side MOSFET gate. This capacitor should be placed as close to the controller as possible and connected between HB and HS.
19	HO	Buck MOSFET gate drive output. Connect to the gate of the high side buck MOSFET through a short, low inductance path.
20	HS	Buck MOSFET source pin. Connect to the source terminal of the high side buck MOSFET and the bootstrap capacitor.
	EP	Solder to the ground plane under the IC to aid in heat dissipation.

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

VIN, EN, VOUT to GND	-0.3V to 76V
VCC, LO, VCCX, UVLO to GND (Note 5)	-0.3 to 15V
HB to HS	-0.3 to 15V
HO to HS	-0.3 to HB+0.3V
HS to GND	-4V to 76V
CSG, CS to GND	-0.3V to +0.3V
RAMP, SS, COMP, FB, SYNC, RT to GND	-0.3 to 7V
ESD Rating HBM (Note 2)	2 kV
Storage Temperature Range	-55°C to +150°C
Junction Temperature	+150°C

Operating Ratings (Note 1)

VIN (Note 4)	3V to 75V
VCC, VCCX	4.75V to 14V
Junction Temperature	-40°C to +125°C

Electrical Characteristics

Limits in standard type are for T_J = 25°C only; limits in **boldface type** apply over the junction temperature range of -40°C to +125°C. Unless otherwise specified, the following conditions apply: VIN = 48V, VCCX = 0V, EN = 5V, RT = 29.11 kΩ, No load on LO and HO (Note 3).

Symbol	Parameter	Conditions	Min	Typ	Max	Units
VIN SUPPLY						
I_{BIAS}	VIN Operating Current	VCCX = 0V		4.5	**5.5**	mA
I_{BIASX}	VIN Operating Current	VCCX = 5V		1	**1.85**	mA
I_{STDBY}	VIN Shutdown Current	EN = 0V		1	**10**	µA
VCC REGULATOR						
$V_{CC(REG)}$	VCC Regulation	VCCX = 0V	**6.8**	7	**7.2**	V
$V_{CC(REG)}$	VCC Regulation	VCCX = 0V, VIN = 6V	**5**	5.25	**5.5**	V
	VCC Sourcing Current Limit	VCC = 0	**21**	35		mA
	VCCX Switch threshold	VCCX Rising	**3.68**	3.85	**4.02**	V
	VCCX Switch hysterisis			0.2		V
	VCCX Switch RDS(ON)	ICCX = 10 mA		5	**12**	Ω
	VCCX Switch Leakage	VCCX = 0V		0.5	**1**	µA
	VCCCX Pull-down Resistance	VCCX = 3V		70		kΩ
	VCC Under-Voltage Lockout Voltage	VCC Rising	**3.52**	3.7	**3.86**	V
	VCC Under-Voltage Hysterisis			0.21		V
	HB DC Bias current	HB-HS = 15V		205	**260**	µA
	VC LDO Mode Turn-off			10		V
EN INPUT						
$V_{IL\ max}$	EN Input Low Threshold				**0.5**	V
$V_{IH\ min}$	EN Input High Threshold		**3.00**			V
	EN Input Bias Current	VEN = 3V	**-1**		**1**	µA
	EN Input Bias Current	VEN = 0.5V	**-1**		**1**	µA
	EN Input Bias Current	VEN = 75V		50		µA
UVLO THRESHOLDS						
	UVLO Standby Threshold	UVLO Rising	**1.191**	1.231	**1.271**	V
	UVLO Threshold Hysteresis			0.105		V
	UVLO Pull-up Current Source	UVLO = 0V		5		µA
	UVLO Pull-down $R_{DS(ON)}$			100	**200**	Ω
SOFT- START						
	SS Current Source	SS = 0V	**7.5**	10.5	**13.5**	µA
	SS to FB Offset	FB = 1.23V		150		mV
	SS Output Low Voltage	Sinking 100 µA, UVLO = 0V		7		mV
ERROR AMPLIFIER						
V_{REF}	FB Reference Voltage	Measured at FB pin, FB = COMP	**1.212**	1.230	**1.248**	V
	FB Input Bias Current	FB = 2V		20	**200**	nA

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	COMP Sink/Source Current		**3**			mA
A_{OL}	DC Gain			80		dB
f_{BW}	Unity Bain Bandwidth			3		MHz
PWM COMPARATORS						
$t_{HO(OFF)}$	Forced HO Off-time		**305**	400	**495**	ns
$T_{ON(MIN)}$	Minimum HO On-time			70		ns
	COMP to Comparator Offset			200		mV
OSCILLATOR (RT PIN)						
f_{SW1}	Frequency 1	RT = 29.11 kΩ	**178**	200	**224**	kHz
f_{SW2}	Frequency 2	RT = 9.525 kΩ	**450**	515	**575**	kHz
SYNC						
	Sync threshold falling			1.3		V
CURRENT LIMIT						
$V_{CS(TH)}$	Cycle-by-cycle Sense Voltage Threshold (CS-CSG)	RAMP = 0 Buck Mode	**-103**	-125	**-147**	mV
$V_{CS(THX)}$	Cycle-by-cycle Sense Voltage Threshold (CS-CSG)	RAMP = 0 Buck-Boost Mode	**-218**	-255	**-300**	mV
	CS Bias Current	CS = 0V		45	**60**	µA
	CSG Bias Current	CSG = 0V		45	**60**	µA
	Current Limit Fault Timer			256		cycles
RAMP GENERATOR						
I_{R1}	RAMP Current 1	VIN = 60V, VOUT = 10V	**245**	305	**365**	µA
I_{R2}	RAMP Current 2	VIN = 12V, VOUT = 12V	**95**	115	**135**	µA
I_{R3}	RAMP Current 3	VIN = 5V, VOUT = 12V	**65**	80	**95**	µA
	VOUT Bias Current	VOUT = 48V		245		µA
LOW SIDE (LO) GATE DRIVER						
V_{OLL}	LO Low-state Output Voltage	I_{LO} = 100 mA	**0.095**	0.14	**0.23**	V
V_{OHL}	LO High-state Output Voltage	I_{LO} = -100 mA $V_{OHL} = V_{CC}-V_{LO}$		0.25		V
	LO Rise Time	C-load = 1 nF, VCC = 8V		16		ns
	LO Fall Time	C-load = 1 nF, VCC = 8V		14		ns
I_{OHL}	Peak LO Source Current	V_{LO} = 0V, VCC = 8V		2.2		A
I_{OLL}	Peak LO Sink Current	V_{LO} = VCC = 8V		2.7		A
HIGH SIDE (HO) GATE DRIVER						
V_{OLH}	HO Low-state Output Voltage	I_{HO} = 100 mA	**0.1**	0.135	**0.21**	V
V_{OHH}	HO High-state Output Voltage	I_{HO} = -100 mA, $V_{OHH} = V_{HB}-V_{OH}$		0.25		V
	HO Rise Time	C-load = 1 nF, VCC = 8V		14		ns
	HO Fall Time	C-load = 1 nF, VCC = 8V		12		ns
I_{OHH}	Peak HO Source Current	V_{HO} = 0V, VCC = 8V		2.2		V
I_{OLH}	Peak HO Sink Current	V_{HO} = VCC = 8V		3.5		V
	HB-HS Under Voltage Lock-out			3		V
BUCK-BOOST CHARACTERISTICS						
	Buck-Boost Mode	Buck Duty Cycle (Note 5)	**69**	75	**80**	%
THERMAL						
T_{SD}	Thermal Shutdown Temp.			165		°C
	Thermal Shutdown Hysterisis			25		°C
θ_{JA}	Junction to Ambient			40		°C/W
θ_{JC}	Junction to Case			4		°C/W

Note 1: Absolute Maximum Ratings are limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is intended to be functional, but does not guarantee specific performance limits. For guaranteed specifications and test conditions see the Electrical Characteristics.

Note 2: The human body model is a 100pF capacitor discharged through a 1.5 kΩ resistor into each pin.

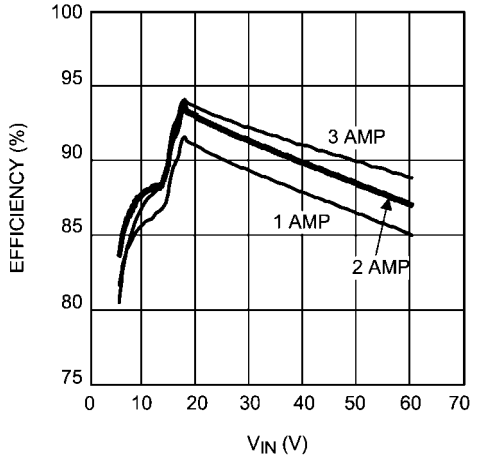
Note 3: Min and Max limits are 100% production tested at 25°C. Limits over the operating temperature range are guaranteed through correlation using Statistical Quality Control (SQC) methods. Limits are used to calculate National's Average Outgoing Quality Level (AOQL).

Note 4: 5V VIN is required to initially start the controller.

Note 5: : When the duty cycle exceeds 75%, the LM5118 controller gradually phases into the Buck-Boost mode.

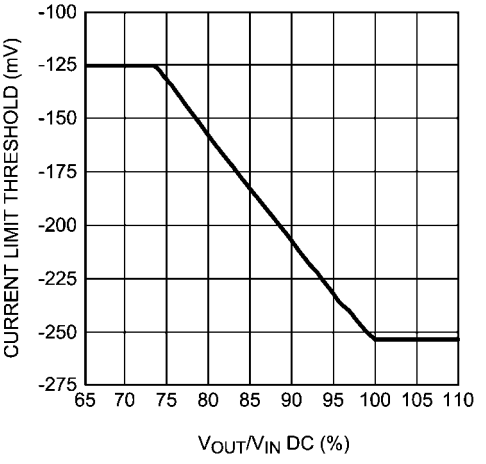
Typical Performance Characteristics

Efficiency vs VIN and IOUT
VOUT = 12V



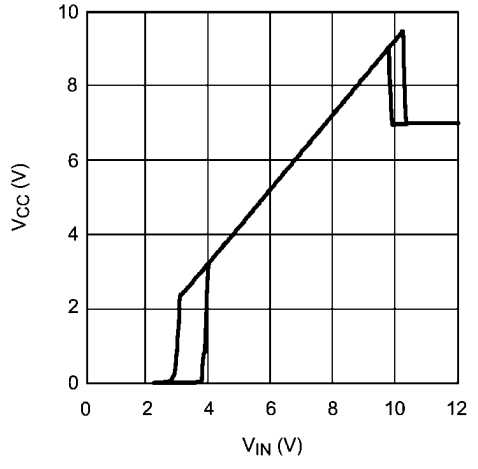
30058503

Current Limit Threshold vs VOUT/VIN
VOUT = 12V



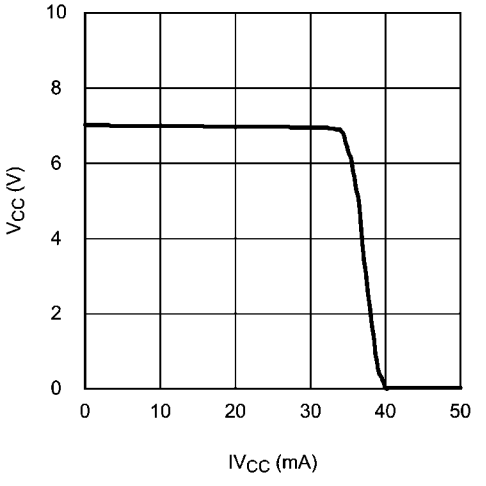
30058504

VCC vs VIN



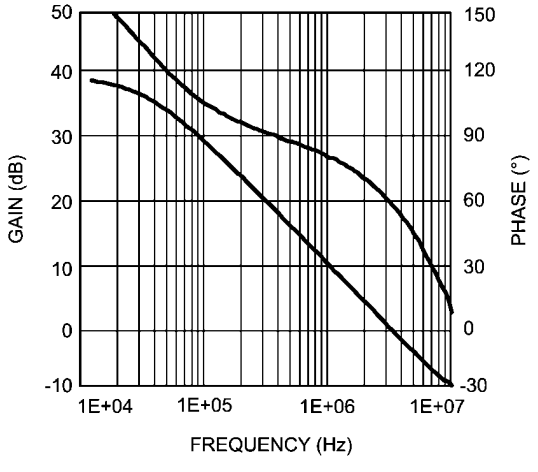
30058505

VCC vs IVCC



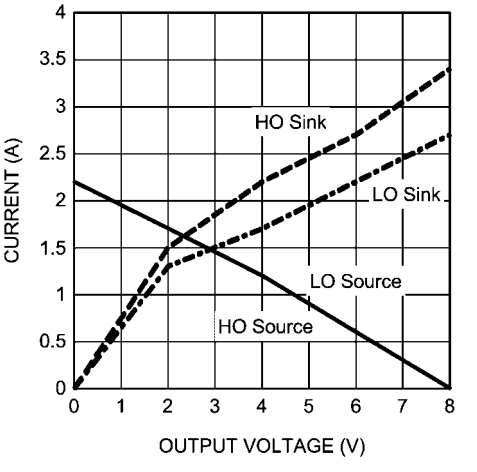
30058506

Error Amplifier Gain/Phase



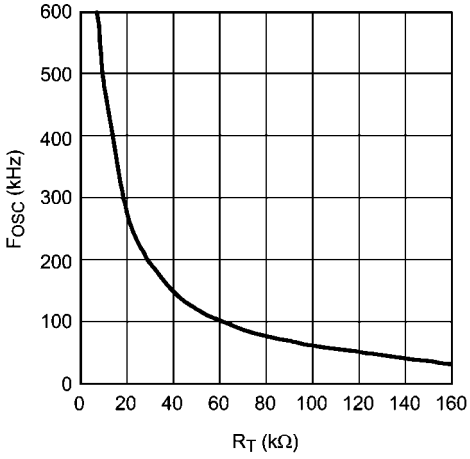
30058507

LO and HO Peak Gate Current vs Output Voltage
VCC = 8V



30058508

Oscillator Frequency vs RT



30058509

Block Diagram and Typical Application Circuit

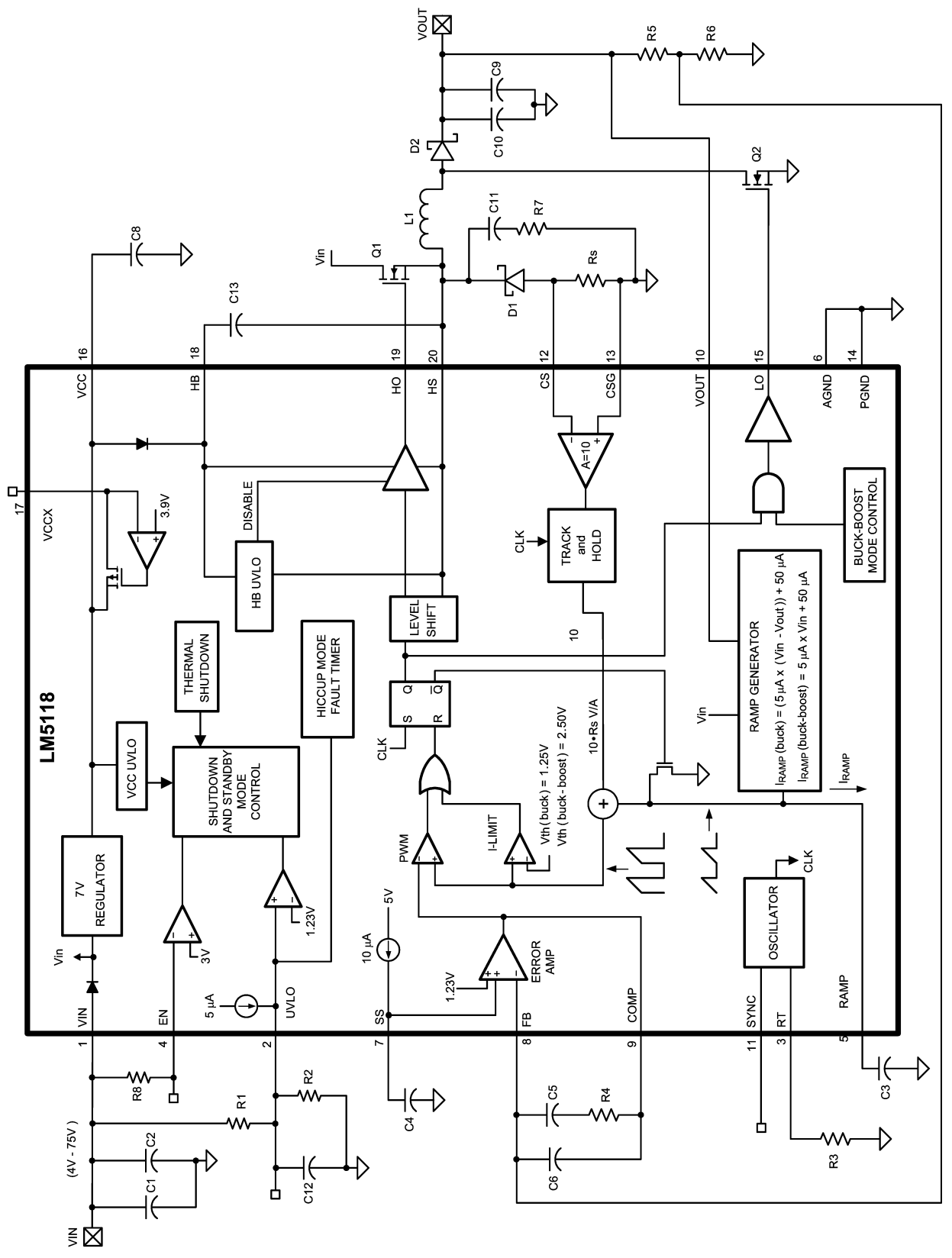


FIGURE 1.

Detailed Operating Description

The LM5118 high voltage switching regulator features all of the functions necessary to implement an efficient high voltage buck or buck-boost regulator using a minimum of external components. The regulator switches smoothly from buck to buck-boost operation as the input voltage approaches the output voltage, allowing operation with the input greater than or less than the output voltage. This easy to use regulator integrates high-side and low-side MOSFET drivers capable of supplying peak currents of 2 Amps. The regulator control method is based on current mode control utilizing an emulated current ramp. Peak current mode control provides inherent line feed-forward, cycle-by-cycle current limiting and ease of loop compensation. The use of an emulated control ramp reduces noise sensitivity of the pulse-width modulation circuit, allowing reliable processing of very small duty cycles necessary in high input voltage applications. The operating frequency is user programmable from 50 kHz to 500 kHz. An oscillator synchronization pin allows multiple LM5118 regulators to self synchronize or be synchronized to an external clock. Fault protection features include current limiting, thermal shutdown and remote shutdown capability. An undervoltage lockout input allows regulator shutdown when the input voltage is below a user selected threshold, and a low state at the enable pin will put the regulator into an extremely low current shutdown state. The device is available in the TSSOP-20EP package featuring an exposed pad to aid in thermal dissipation.

A buck-boost regulator can maintain regulation for input voltages either higher or lower than the output voltage. The challenge is that buck-boost power converters are not as efficient as buck regulators. The LM5118 has been designed as a dual mode controller whereby the power converter acts as a buck regulator while the input voltage is above the output. As the input voltage approaches the output voltage, a gradual transition to the buck-boost mode occurs. The dual mode approach maintains regulation over a wide range of input voltages, while maintaining the optimal conversion efficiency in the normal buck mode. The gradual transition between modes eliminates disturbances at the output during transitions. *Figure 2* shows the basic operation of the LM5118 regulator in the buck mode. In buck mode, transistor Q1 is active and Q2 is disabled. The inductor current ramps in proportion to the Vin - Vout voltage difference when Q1 is active and ramps down through the re-circulating diode D1 when Q1 is off. The first order buck mode transfer function is VOUT/VIN = D, where D is the duty cycle of the buck switch, Q1.

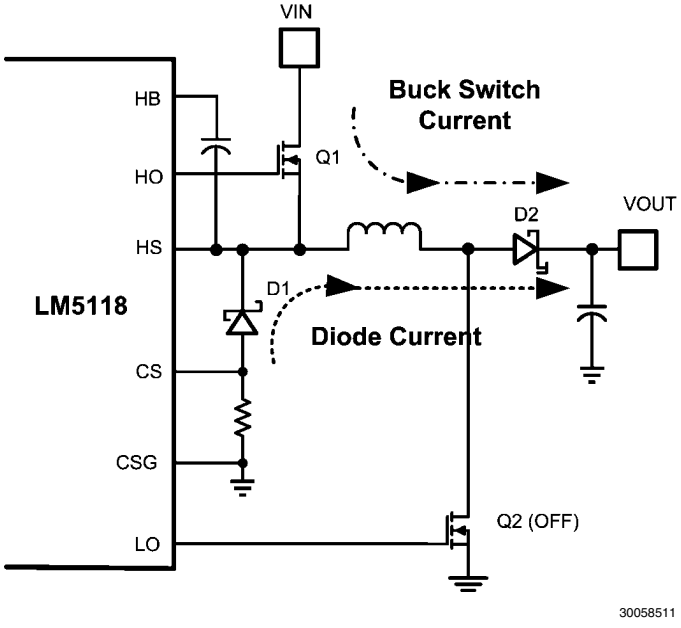


FIGURE 2. Buck Mode Operation

Figure 3 shows the basic operation of buck-boost mode. In buck-boost mode both Q1 and Q2 are active for the same time interval each cycle. The inductor current ramps up (proportional to VIN) when Q1 and Q2 are active and ramps down through the re-circulating diode during the off time. The first order buck-boost transfer function is VOUT/VIN = D/(1-D), where D is the duty cycle of Q1 and Q2.

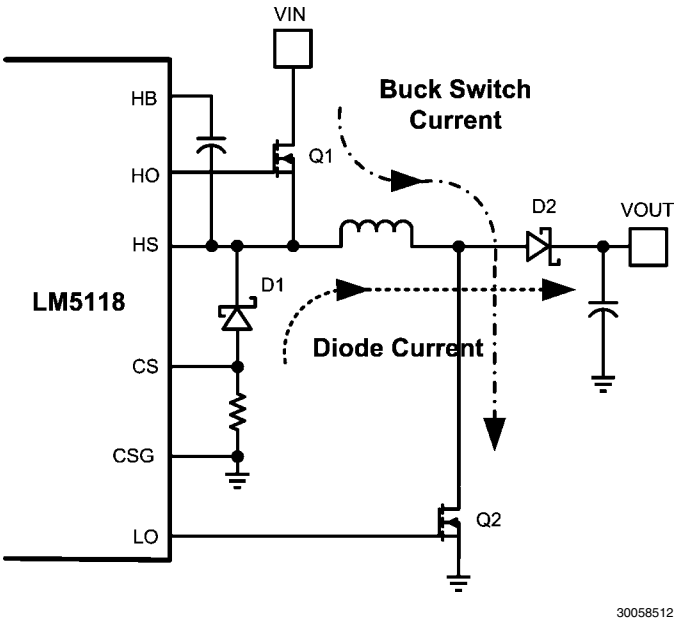


FIGURE 3. Buck-Boost Mode Operation

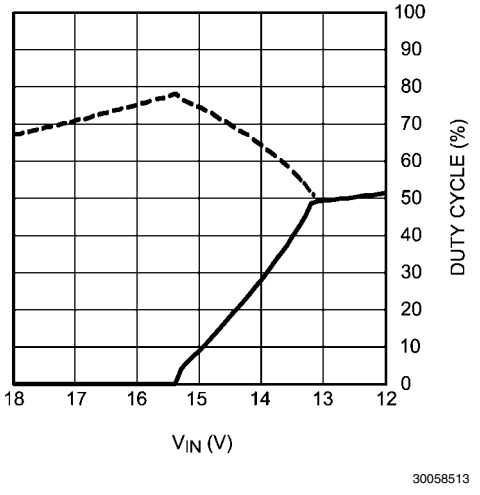


FIGURE 4. Mode Dependence on Duty Cycle (VOUT =12V)

Operation Modes

Figure 4 illustrates how duty cycle affects the operational mode and is useful for reference in the following discussions. Initially, only the buck switch is active and the buck duty cycle increases to maintain output regulation as VIN decreases. When VIN is approximately equal to 15.5V, the boost switch begins to operate with a low duty cycle. If VIN continues to fall, the boost switch duty cycle increases and the buck switch duty cycle decreases until they become equal at VIN = 13.2V.

Buck Mode Operation: VIN > VOUT

The LM5118 buck-boost regulator operates as a conventional buck regulator with emulated current mode control while VIN is greater than VOUT and the buck mode duty cycle is less than 75%. In buck mode, the LO gate drive output to the boost switch remains low.

Buck-Boost Mode Operation: VIN ≅ VOUT

When VIN decreases relative to VOUT, the duty cycle of the buck switch will increase to maintain regulation. Once the duty cycle reaches 75%, the boost switch starts to operate with a very small duty cycle. As VIN is further decreased, the boost switch duty cycle increases until it is the same as the buck switch. As VIN is further decreased below VOUT, the buck and boost switch operate together with the same duty cycle and the regulator is in full buck-boost mode. This feature allows the regulator to transition smoothly from buck to buck-boost mode. It should be noted that the regulator can be designed to operate with VIN less than 4 volts, but VIN must be at least 5 volts during start-up. *Figure 5* presents a timing illustration of the gradual transition from buck to buck-boost mode when the input voltage ramps downward over a few switching cycles.

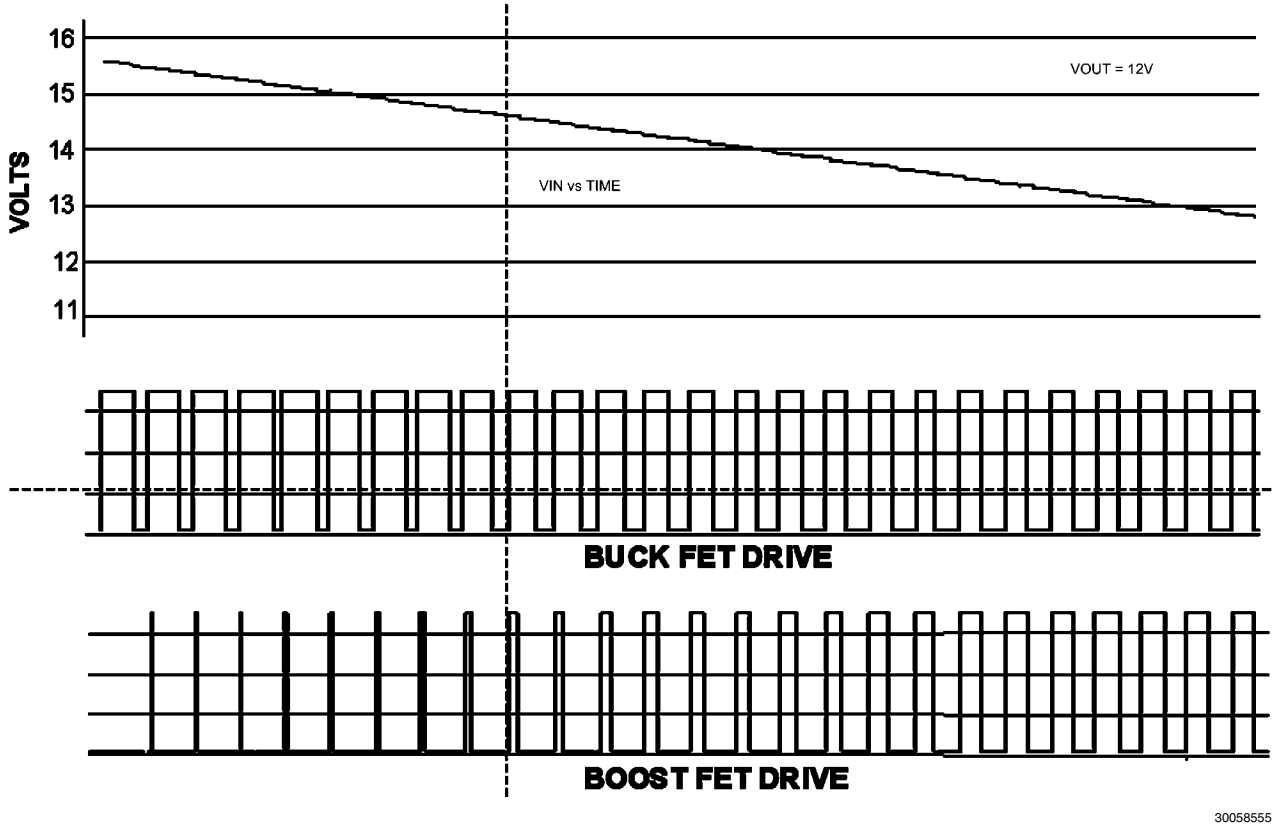


FIGURE 5. Buck (HO) and Boost (LO) Switch Duty Cycle vs. Time, Illustrating Gradual Mode Change with Decreasing Input Voltage

High Voltage Start-Up Regulator

The LM5118 contains a dual mode, high voltage linear regulator that provides the VCC bias supply for the PWM controller and the MOSFET gate driver. The VIN input pin can be connected directly to input voltages as high as 75V. For input voltages below 10V, an internal low dropout switch connects VCC directly to VIN. In this supply range, VCC is approximately equal to VIN. For VIN voltages greater than 10V, the low dropout switch is disabled and the VCC regulator is enabled to maintain VCC at approximately 7V. A wide operating range of 4V to 75V (with a startup requirement of at least 5 volts) is achieved through the use of this dual mode regulator.

The output of the VCC regulator is current limited to 35 mA, typical. Upon power up, the regulator sources current into the capacitor connected to the VCC pin. When the voltage at the VCC pin exceeds the VCC under-voltage threshold of 3.7V and the UVLO input pin voltage is greater than 1.23V, the gate driver outputs are enabled and a soft-start sequence begins. The gate driver outputs remain enabled until VCC falls below 3.5V or the voltage at the UVLO pin falls below 1.13V.

In many applications the regulated output voltage or an auxiliary supply voltage can be applied to the VCCX pin to reduce the IC power dissipation. For output voltages between 4V and 15V, VOUT can be connected directly to VCCX. When the voltage at the VCCX pin is greater than 3.85V, the internal VCC regulator is disabled and an internal switch connects VCCX to VCC, reducing the internal power dissipation.

In high voltage applications extra care should be taken to ensure the VIN pin voltage does not exceed the absolute maximum voltage rating of 76V. During line or load transients, voltage ringing on the VIN line that exceeds the absolute maximum rating can damage the IC. Both careful PC board layout and the use of quality bypass capacitors located close to the VIN and GND pins are essential.

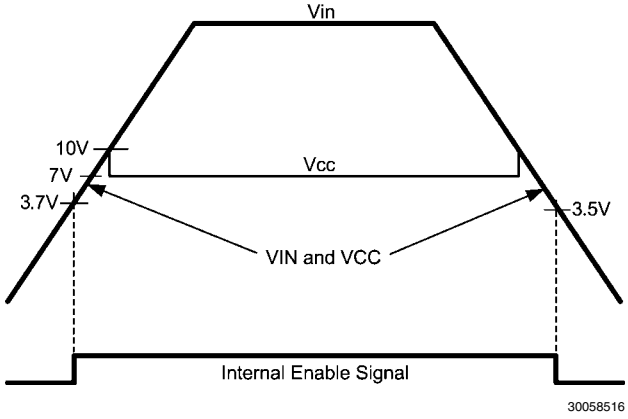


FIGURE 6. VIN and VCC Sequencing

Enable

The LM5118 contains an enable function which provides a very low input current shutdown mode. If the EN pin is pulled below 0.5V, the regulator enters shutdown mode, drawing less than 10 µA from the VIN pin. Raising the EN input above

3V returns the regulator to normal operation. The EN pin can be tied directly to the VIN pin if this function is not needed. It must not be left floating. A 1 MΩ pull-up resistor to VIN can be used to interface with an open collector or open drain control signal.

UVLO

An under-voltage lockout pin is provided to disable the regulator when the input is below the desired operating range. If the UVLO pin is below 1.13V, the regulator enters a standby mode with the outputs disabled, but with VCC regulator operating. If the UVLO input exceeds 1.23V, the regulator will resume normal operation. A voltage divider from the input to ground can be used to set a VIN threshold to disable the regulator in brown-out conditions or for low input faults.

If a current limit fault exists for more than 256 clock cycles, the regulator will enter a "hiccup" mode of current limiting and the UVLO pin will be pulled low by an internal switch. This switch turns off when the UVLO pin approaches ground potential allowing the UVLO pin to rise. A capacitor connected to the UVLO pin will delay the return to a normal operating level and thereby set the off-time of the hiccup mode fault protection. An internal 5 µA pull-up current pulls the UVLO pin to a high state to ensure normal operation when the VIN UVLO function is not required and the pin is left floating.

Oscillator and Sync Capability

The LM5118 oscillator frequency is set by a single external resistor connected between the RT pin and the AGND pin. The R_T resistor should be located very close to the device and connected directly to the pins of the IC. To set a desired oscillator frequency (f), the necessary value for the R_T resistor can be calculated from the following equation:

$$R_T = \frac{6.4 \times 10^9}{f} - 3.02 \times 10^3$$

The SYNC pin can be used to synchronize the internal oscillator to an external clock. The external clock must be of higher frequency than the free-running frequency set by the R_T resistor. A clock circuit with an open drain output is the recommended interface from the external clock to the SYNC pin. The clock pulse duration should be greater than 15 ns.

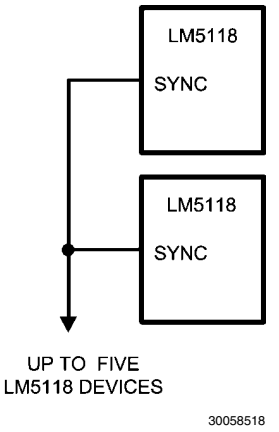


FIGURE 7. Sync from Multiple Devices

Multiple LM5118 devices can be synchronized together simply by connecting the SYNC pins together. In this configuration all of the devices will be synchronized to the highest frequency device. The diagram in *Figure 7* illustrates the SYNC input/output features of the LM5118. The internal oscillator circuit drives the SYNC pin with a strong pull-down/ weak pull-up inverter. When the SYNC pin is pulled low, either by the internal oscillator or an external clock, the ramp cycle of the oscillator is terminated and forced 400 ns off-time is initiated before a new oscillator cycle begins. If the SYNC pins of several LM5118 IC's are connected together, the IC with the highest internal clock frequency will pull all the connected SYNC pins low and terminate the oscillator ramp cycles of the other IC's. The LM5118 with the highest programmed clock frequency will serve as the master and control the switching frequency of all the devices with lower oscillator frequencies.

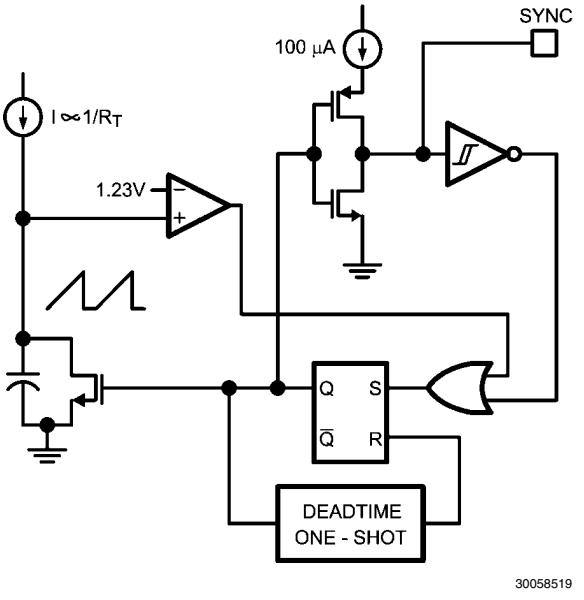


FIGURE 8. Simplified Oscillator and Block Diagram with Sync I/O Circuit

Error Amplifier and PWM Comparator

The internal high gain error amplifier generates an error signal proportional to the difference between the regulated output voltage and an internal precision reference (1.23V). The output of the error amplifier is connected to the COMP pin. Loop compensation components, typically a type II network illustrated in *Figure 1* are connected between the COMP and FB pins. This network creates a low frequency pole, a zero, and a noise reducing high frequency pole. The PWM comparator compares the emulated current sense signal from the RAMP generator to the error amplifier output voltage at the COMP pin. The same error amplifier is used for operation in buck and buck-boost mode.

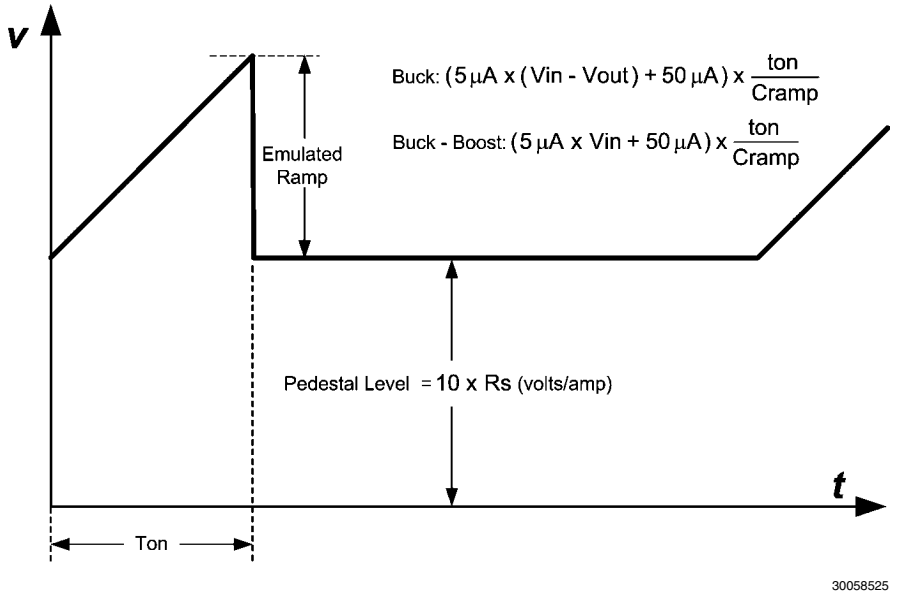

v
Buck: (5 µA x (Vin - Vout) + 50 µA) x ton/Cramp
Buck - Boost: (5 µA x Vin + 50 µA) x ton/Cramp
Emulated Ramp
Pedestal Level = 10 x Rs (volts/amp)
t
Ton
30058525

FIGURE 9. Composition of Emulated Current Signal

Ramp Generator

The ramp signal of a pulse-width modulator with current mode control is typically derived directly from the buck switch drain current. This switch current corresponds to the positive slope portion of the inductor current signal. Using this signal for the PWM ramp simplifies the control loop transfer function to a single pole response and provides inherent input voltage feed-forward compensation. The disadvantage of using the buck switch current signal for PWM control is the large leading edge spike due to circuit parasitics. The leading edge spike must be filtered or blanked to avoid early termination of the PWM pulse. Also, the current measurement may introduce significant propagation delays. The filtering, blanking time and propagation delay limit the minimal achievable pulse width. In applications where the input voltage may be relatively large in comparison to the output voltage, controlling a small pulse width is necessary for regulation. The LM5118 utilizes a unique ramp generator which does not actually measure the buck switch current but instead creates a signal representing or emulating the inductor current. The emulated ramp provides signal to the PWM comparator that is free of leading edge spikes and measurement or filtering delays. The current reconstruction is comprised of two elements, a sample-and-hold pedestal level and a ramp capacitor which is charged by a controlled current source. Refer to *Figure 9* for details.

The sample-and-hold pedestal level is derived from a measurement of the re-circulating current through a current sense resistor in series with the re-circulating diode of the buck regulator stage. A small value current sensing resistor is required between the re-circulating diode anode and ground. The CS and CSG pins should be Kelvin connected directly to the sense resistor. The voltage level across the sense resistor is sampled and held just prior to the onset of the next conduction interval of the buck switch. The current sensing and sample-and-hold provide the DC level of the reconstructed current signal. The sample and hold of the re-circulating diode current is valid for both buck and buck-boost modes. The positive slope inductor current ramp is emulated by an external capacitor connected from the RAMP pin to the AGND and an internal voltage controlled current source. In buck mode, the ramp current source that emulates the inductor current is a function of the VIN and VOUT voltages per the following equation:

$$I_{RAMP}\ (buck) = \frac{5\ \mu A}{V} \times (VIN - VOUT) + 50\ \mu A$$

In buck-boost mode, the ramp current source is a function of the input voltage VIN, per the following equation:

$$I_{RAMP}\ (buck - boost) = \frac{5\ \mu A}{V} \times VIN + 50\ \mu A$$

Proper selection of the RAMP capacitor (C_{RAMP}) depends upon the value of the output inductor (L) and the current sense resistor (R_S). For proper current emulation, the sample and hold pedestal value and the ramp amplitude must have the same relative relationship to the actual inductor current. That is:

$$R_S \times A = \frac{g_m \times L}{C_{RAMP}}$$

$$C_{RAMP} = \frac{g_m \times L}{A \times R_S}$$

Where g_m is the ramp generator transconductance (5 µA/V) and A is the current sense amplifier gain (10V/V). The ramp capacitor should be located very close to the device and connected directly to the RAMP and AGND pins.

The relationship between the average inductor current and the pedestal value of the sampled inductor current can cause instability in certain operating conditions. This instability is known as sub-harmonic oscillation, which occurs when the inductor ripple current does not return to its initial value by the start of the next switching cycle. Sub-harmonic oscillation is normally characterized by observing alternating wide and narrow pulses at the switch node. Adding a fixed slope voltage ramp (slope compensation) to the current sense signal prevents this oscillation. The 50µA of offset current provided from the emulated current source adds enough slope compensation to the ramp signal for output voltages less than or equal to 12V. For higher output voltages, additional slope compensation may be required. In such applications, the ramp capacitor can be decreased from the nominal calculated value to increase the ramp slope compensation.

The pedestal current sample is obtained from the current sense resistor (Rs) connected to the CS and CSG pins. It is sometimes helpful to adjust the internal current sense amplifier gain (A) to a lower value in order to obtain the higher current limit threshold. Adding a pair of external resistors RG in a series with CS and CSG as shown in *Figure 10* reduces the current sense amplifier gain A according to the following equation:

$$A = \frac{10k}{1k + R_G}$$

Current Limit

In the buck mode the average inductor current is equal to the output current (Iout). In buck-boost mode the average inductor current is approximately equal to:

$$Iout \times \left(1 + \frac{VOUT}{VIN}\right)$$

Consequently, the inductor current in buck-boost mode is much larger especially when VOUT is large relative to VIN. The LM5118 provides a current monitoring scheme to protect the circuit from possible over-current conditions. When set correctly, the emulated current sense signal is proportional to the buck switch current with a scale factor determined by the current sense resistor. The emulated ramp signal is applied to the current limit comparator. If the peak of the emulated ramp signal exceeds 1.25V when operating in the buck mode, the PWM cycle is immediately terminated (cycle-by-cycle current limiting). In buck-boost mode the current limit threshold is increased to 2.50V to allow higher peak inductor current. To further protect the external switches during prolonged overload conditions, an internal counter detects consecutive cycles of current limiting. If the counter detects 256 consecutive current limited PWM cycles, the LM5118 enters a low power dissipation hiccup mode. In the hiccup mode, the output drivers are disabled, the UVLO pin is momentarily pulled low, and the soft-start capacitor is discharged. The regulator is restarted with a normal soft-start sequence once the UVLO pin charges back to 1.23V. The hiccup mode off-time can be programmed by an external capacitor connected from UVLO pin to ground. This hiccup cycle will repeat until the output overload condition is removed.

In applications with low output inductance and high input voltage, the switch current may overshoot due to the propagation delay of the current limit comparator and control circuitry. If an overshoot should occur, the sample-and-hold circuit will detect the excess re-circulating diode current. If the sample-and-hold pedestal level exceeds the internal current limit threshold, the buck switch will be disabled and will skip PWM cycles until the inductor current has decayed below the current limit threshold. This approach prevents current runaway conditions due to propagation delays or inductor saturation since the inductor current is forced to decay before the buck switch is turned on again.

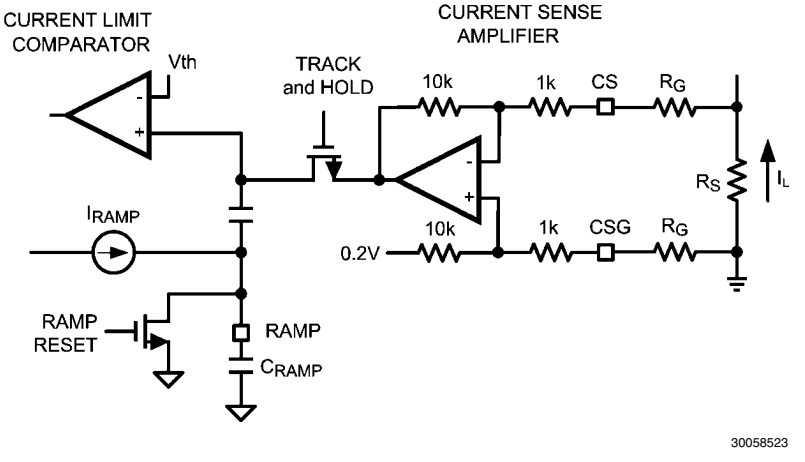


FIGURE 10. Current Limit and Ramp Circuit

Maximum Duty Cycle

Each conduction cycle of the buck switch is followed by a forced minimum off-time of 400ns to allow sufficient time for the re-circulating diode current to be sampled. This forced off-time limits the maximum duty cycle of the controller. The actual maximum duty cycle will vary with the operating frequency as follows:

$$D_{MAX} = 1 - f \times 400 \times 10^{-9}$$

where f is the oscillator frequency in Hz

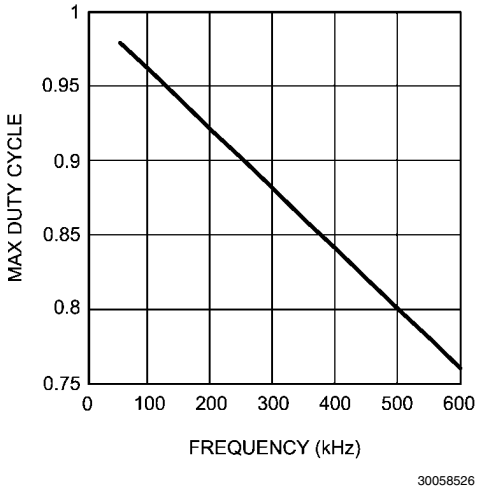


FIGURE 11. Maximum Duty Cycle vs Frequency

Limiting the maximum duty cycle will limit the maximum boost ratio (VOUT/VIN) while operating in buck-boost mode. For example, from *Figure 11*, at an operating frequency of 500 kHz, D_{MAX} is 80%. Using the buck-boost transfer function.

$$D = \frac{Vout}{Vin + Vout}$$

With D= 80%, solving for VOUT results in,

VOUT = 4 x VIN

With a minimum input voltage of 5 volts, the maximum possible output voltage is 20 volts at f = 500 kHz. The buck-boost step-up ratio can be increased by reducing the operating frequency which increases the maximum duty cycle.

Soft-Start

The soft-start feature allows the regulator to gradually reach the initial steady state operating point, thus reducing start-up stresses and surges. The internal 10 µA soft-start current source gradually charges an external soft-start capacitor connected to the SS pin. The SS pin is connected to the positive input of the internal error amplifier. The error amplifier controls the pulse-width modulator such that the FB pin approximately equals the SS pin as the SS capacitor is charged. Once the SS pin voltage exceeds the internal 1.23V reference voltage, the error amp is controlled by the reference instead of the SS pin. The SS pin voltage is clamped by an internal amplifier at a level of 150 mV above the FB pin voltage. This feature provides a soft-start controlled recovery in the event a severe overload pulls the output voltage (and FB pin) well below normal regulation but doesn't persist for 256 clock cycles.

Various sequencing and tracking schemes can be implemented using external circuits that limit or clamp the voltage level of the SS pin. The SS pin acts as a non-inverting input to the error amplifier anytime SS voltage is less than the 1.23V reference. In the event a fault is detected (over-temperature, VCC under-voltage, hiccup current limit), the soft-start capacitor will be discharged. When the fault condition is no longer present, a new soft-start sequence will begin.

HO Ouput

The LM5118 contains a high side, high current gate driver and associated high voltage level shift. This gate driver circuit works in conjunction with an internal diode and an external bootstrap capacitor. A 0.1 µF ceramic capacitor, connected with short traces between the HB pin and HS pin is recommended for most circuit configurations. The size of the bootstrap capacitor depends on the gate charge of the external FET. During the off time of the buck switch, the HS pin voltage is approximately -0.5V and the bootstrap capacitor is charged from VCC through the internal bootstrap diode. When operating with a high PWM duty cycle, the buck switch will be forced off each cycle for 400ns to ensure that the bootstrap capacitor is recharged.

Thermal Protection

Internal Thermal Shutdown circuitry is provided to protect the integrated circuit in the event the maximum junction temperature is exceeded. When activated, typically at 165°C, the controller is forced into a low power reset state, disabling the output driver and the bias regulator. This protection is provided to prevent catastrophic failures from accidental device overheating.

Application Information

The procedure for calculating the external components is illustrated with the following design example. The designations used in the design example correlate to the final schematic shown in *Figure 18*. The design specifications are:

- VOUT = 12V
- VIN = 5V to 75V
- F = 300 kHz
- Minimum load current (CCM operation) = 600 mA
- Maximum load current = 3A

R7 = R_T

RT sets the oscillator switching frequency. Generally speaking, higher operating frequency applications will use smaller components, but have higher switching losses. An operating frequency of 300 kHz was selected for this example as a reasonable compromise for both component size and efficiency. The value of R_T can be calculated as follows:

$$R_T = \frac{6.4 \times 10^9}{f} - 3.02 \times 10^3$$

therefore, R7 = 18.3 kΩ

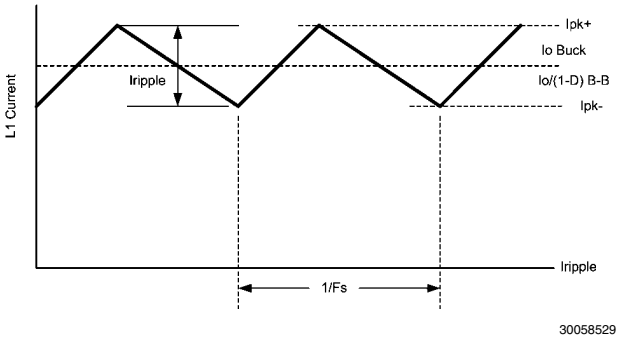


FIGURE 12. Inductor Current Waveform

INDUCTOR SELECTION

L1

The inductor value is determined based upon the operating frequency, load current, ripple current and the input and output voltages. Refer to *Figure 12* for details.

To keep the circuit in continuous conduction mode (CCM), the maximum ripple current IRIPPLE should be less than twice the minimum load current. For the specified minimum load of 0.6A, the maximum ripple current is 1.2A p-p. Also, the minimum value of L must be calculated both for a buck and buck-boost configurations. The final value of inductance will generally be a compromise between the two modes. It is desirable to have a larger value inductor for buck mode, but the saturation current rating for the inductor must be large for buck-boost mode, resulting in a physically large inductor. Additionally, large value inductors present buck-boost mode loop compensation challenges which will be discussed in error amplifier configuration section. For the design example, the inductor values in both modes are calculated as:

$$L1 = \frac{V_{OUT}\,(V_{IN1} - V_{OUT})}{V_{IN1} \times f \times I_{RIPPLE}} \quad \text{Buck Mode}$$

$$L1 = \frac{V_{IN2}\,(V_{OUT})}{(V_{OUT} + V_{IN2}) \times f \times I_{RIPPLE}} \quad \text{Buck-Boost Mode}$$

Where:

V_{OUT} is the output voltage

V_{IN1} is the maximum input voltage

f is the switching frequency

I_{RIPPLE} is the selected inductor peak to peak ripple current (1.2 A selected for this example)

V_{IN2} is the minimum input voltage

The resulting inductor values are:

L1 = 28 µH, Buck Mode

L1 = 9.8 µH Buck-Boost mode

A 10 µH inductor was selected which is a compromise between these values, while favoring the buck-boost mode. As will be illustrated in the compensation section below, the inductor value should be as low as possible to move the buck-boost right-half-plane zero to a higher frequency. The ripple current is then rechecked with the selected inductor value using the equations above,

$I_{RIPPLE(BUCK)}$ = 3.36A

$I_{RIPPLE(BUCK\text{-}BOOST)}$ = 1.17A

Because the inductor selected is lower than calculated for the Buck mode, the minimum load current for CCM in buck mode is 1.68 A at maximum VIN.

With a 10 µH inductor, the worst case peak inductor currents can be estimated for each case, assuming a 20% inductor value tolerance.

$$I_{1(PEAK)} = \frac{I_{OUT}}{0.8} + \frac{I_{RIPPLE}}{2} \quad \text{Buck Mode}$$

$$I_{2(PEAK)} = \frac{(V_{OUT} + Vin)I_{OUT}}{0.8 \times V_{IN2}} + \frac{I_{RIPPLE}}{2} \quad \text{Buck-Boost Mode}$$

For this example, the two equations yield:

$I_{1(PEAK)}$ = 5.43A

$I_{2(PEAK)}$ = 13.34A

An acceptable current limit setting would be 6.7A for buck mode since the LM5118 automatically doubles the current limit threshold in buck-boost mode. The selected inductor must have a saturation current rating at least as high as the buck-boost mode cycle-by-cycle current limit threshold, in this case at least 13.5A. A 10 µH 15 amp inductor was chosen for this application.

R13 = R_{SENSE}

To select the current sense resistor value, begin by calculating the value of R_{SENSE} for both modes of operation.

$$R13_{(BUCK)} = \frac{1.25V}{10 \times I_{PEAK}}$$

$R13_{(BUCK)}$ = 23 mΩ

For the buck-boost mode, R_{SENSE} is given by:

$$R13_{(BUCK\text{-}BOOST)} = \frac{2.5V}{10 \times I_{PEAK}}$$

$R13_{(BUCK\text{-}BOOST)}$ = 18.7 mΩ

A R_{SENSE} value of no more than 18.7 mΩ must be used to guarantee the required maximum output current in the buck-boost mode. A value of 15 mΩ was selected for component tolerances and is a standard value.

R13 = 15 mΩ

C15 = C_{RAMP}

With the inductor value selected, the value of C3 necessary for the emulation ramp circuit is:

$$C15 = C_{RAMP} = \frac{L \times 10^{-6}}{2 \times R_{SENSE}}$$

With the inductance value (L1) selected as 10 µH, the calculated value for C_{RAMP} is 333 pF. A standard value of 330 pF was selected.

C9 - C12 = OUTPUT CAPACITORS

In buck-boost mode, the output capacitors C9 - C12 must supply the entire output current during the switch on-time. For this reason, the output capacitors are chosen for operation in buck-boost mode, the demands being much less in buck operation. Both bulk capacitance and ESR must be considered

to guarantee a given output ripple voltage. Buck-boost mode capacitance can be estimated from:

$$C_{MIN} = \frac{I_{OUT} \times D_{MAX}}{f \times \Delta V_{OUT}} \text{ With } D_{MAX} = \frac{V_{OUT}}{V_{IN2} + V_{OUT}}$$

ESR requirements can be estimated from:

$$ESR_{MAX} = \frac{\Delta V_{OUT}}{I_{PEAK}}$$

For our example, with a ΔVOUT (output ripple) of 50 mV,

C_{MIN} = 141 µF

ESR_{MAX} = 3.8 mΩ

If hold-up times are a consideration, the values of input/output capacitors must be increased appropriately. Note that it is usually advantageous to use multiple capacitors in parallel to achieve the ESR value required. Also, it is good practice to put a .1 µF - .47 µF ceramic capacitor directly on the output pins of the supply to reduce high frequency noise. Ceramic capacitors have good ESR characteristics, and are a good choice for input and output capacitors. It should be noted that the effective capacitance of ceramic capacitors decreases with dc bias. For larger bulk values of capacitance, a low ESR electrolytic is usually used. However, electrolytic capacitors have poor tolerance, especially over temperature, and the selected value should be selected larger than the calculated value to allow for temperature variation. Allowing for component tolerances, the following values of Cout were chosen for this design example:

Two 180 µF Oscon electrolytic capacitors for bulk capacitance

Two 47 µF ceramic capacitors to reduce ESR

Two 0.47 µF ceramic capacitors to reduce spikes at the output .

D1

Reverse recovery currents degrade performance and decrease efficiency. For these reasons, a Schottky diode of appropriate ratings should be used for D1. The voltage rating of the boost diode should be equal to VOUT plus some margin. Since D1 only conducts during the buck switch off time in either mode, the current rating required is:

$I_{DIODE} = I_{OUT}$ x (1-D) Buck Mode

$I_{DIODE} = I_{OUT}$ Buck-Boost Mode

D4

A Schottky type re-circulating diode is required for all LM5118 applications. The near ideal reverse recovery characteristics and low forward voltage drop are particularly important diode characteristics for high input voltage and low output voltage applications. The reverse recovery characteristic determines how long the current surge lasts each cycle when the buck switch is turned on. The reverse recovery characteristics of Schottky diodes minimize the peak instantaneous power in the buck switch during the turn-on transition. The reverse breakdown rating of the diode should be selected for the maximum VIN plus some safety margin.

The forward voltage drop has a significant impact on the conversion efficiency, especially for applications with a low output voltage. “Rated” current for diodes vary widely from various manufacturers. For the LM5118 this current is user selectable through the current sense resistor value. Assuming a worst case 0.6V drop across the diode, the maximum diode power dissipation can be high. The diode should have a voltage rating of VIN and a current rating of IOUT. A conservative design would at least double the advertised diode rating since specifications between manufacturers vary. For the reference design a 100V, 10A Schottky in a D2PAK package was selected.

C1 - C5 = INPUT CAPACITORS

A typical regulator supply voltage has a large source impedance at the switching frequency. Good quality input capacitors are necessary to limit the ripple voltage at the VIN pin while supplying most of the switch current during the buck switch on-time. When the buck switch turns on, the current into the buck switch steps from zero to the lower peak of the inductor current waveform, then ramps up to the peak value, and then drops to the zero at turn-off. The RMS current rating of the input capacitors depends on which mode of operation is most critical.

$$I_{RMS(BUCK)} = I_{OUT} \sqrt{D(1 - D)}$$

This value is a maximum at 50% duty cycle which corresponds to VIN = 75 volts.

$$I_{RMS(BUCK\text{-}BOOST)} = \frac{I_{OUT}}{1 - D} \sqrt{D(1 - D)}$$

Checking both modes of operation we find:

$I_{RMS(BUCK)}$ = 1.5 Amps

$I_{RMS(BUCK\text{-}BOOST)}$ = 4.7 Amps

Therefore C1 - C5 should be sized to handle 4.7A of ripple current. Quality ceramic capacitors with a low ESR should be selected. To allow for capacitor tolerances, four 2.2 µF, 100V ceramic capacitors will be used. If step input voltage transients are expected near the maximum rating of the LM5118, a careful evaluation of the ringing and possible spikes at the device VIN pin should be completed. An additional damping network or input voltage clamp may be required in these cases.

C20

The capacitor at the VCC pin provides noise filtering and stability for the VCC regulator. The recommended value of C20 should be no smaller than 0.1 µF, and should be a good quality, low ESR, ceramic capacitor. A value of 1 µF was selected for this design. C20 should be 10 x C8.

If operating without VCCX, then

$$f_{OSC} \times (Q_C \text{Buck + Boost}) + I_{LOAD(INTERNAL)}$$

must be less than the VCC current limit.

C8

The bootstrap capacitor between the HB and HS pins supplies the gate current to charge the buck switch gate at turn-on. The recommended value of C8 is 0.1 µF to 0.47 µF, and should be a good quality, low ESR, ceramic capacitor. A value of 0.1 µF was chosen for this design.

C16 = C_{SS}

The capacitor at the SS pin determines the soft-start time, i.e. the time for the reference voltage and the output voltage, to reach the final regulated value. The time is determined from:

$$t_{SS} = \frac{C16 \times 1.23V}{10\ \mu A}$$

and assumes a current limit>Iload + ICout

For this application, a C16 value of 0.1 µF was chosen which corresponds to a soft-start time of about 12 ms.

R8, R9

R8 and R9 set the output voltage level, the ratio of these resistors is calculated from:

$$\frac{R8}{R9} = \frac{V_{OUT}}{1.23V} - 1$$

For a 12V output, the R8/R9 ratio calculates to 9.76. The resistors should be chosen from standard value resistors and a good starting point is to select resistors within power ratings appropriate for the output voltage. Values of 309Ω for R9 and 2.67 kΩ for R8 were selected.

R1, R3, C21

A voltage divider can be connected to the UVLO pin to set a minimum operating voltage $VIN_{(UVLO)}$ for the regulator. If this feature is required, the easiest approach to select the divider resistor values is to choose a value for R1 between 10 kΩ and 100 kΩ, while observing the minimum value of R1 necessary to allow the UVLO switch to pull the UVLO pin low. This value is:

$R1 \geq 1000 \times V_{IN(MAX)}$

R1 ≥ 75k in our example

R3 is then calculated from

$$R3 = 1.23 \times \left[\frac{R1}{V_{IN(MIN)} + 5\ \mu A \times R1 - 1.23}\right]$$

Since $VIN_{(MIN)}$ for our example is 5V, set $VIN_{(UVLO)}$ to 4.0V for some margin in component tolerances and input ripple.

R1 = 75k is chosen since it is a standard value

R3 = 29.332k is calculated from the equation above. 29.4k was used since it is a standard value.

Capacitor C21 provides filtering for the divider and the off time of the "hiccup" duty cycle during current limit. The voltage at the UVLO pin should never exceed 15V when using an external set-point divider. It may be necessary to clamp the UVLO pin at high input voltages.

Knowing the desired off time during "hiccup" current limit, the value of C21 is given by:

$$t_{OFF} = -\left(\frac{R1 \times R3}{R1 + R3}\right) \times C21 \times \ln\left[1 - \frac{1.23 \times (R1 + R3)}{V_{IN} \times R1}\right]$$

Notice that t_{OFF} varies with V_{IN}

In this example, C21 was chosen to be 0.1 µF. This will set the t_{OFF} time to 956 µs with VIN = 12V.

R2

A 1M pull-up resistor connected from the EN pin to the VIN pin is sufficient to keep enable in a high state if on-off control is not used.

SNUBBER

A snubber network across the buck re-circulating diode reduces ringing and spikes at the switching node. Excessive ringing and spikes can cause erratic operation and increase noise at the regulator output. In the limit, spikes beyond the maximum voltage rating of the LM5118 or the re-circulating diode can damage these devices. Selecting the values for the snubber is best accomplished through empirical methods. First, make sure the lead lengths for the snubber connections are very short. Start with a resistor value between 5 and 20 Ohms. Increasing the value of the snubber capacitor results in more damping, however the snubber losses increase. Select a minimum value of the capacitor that provides adequate clamping of the diode waveform at maximum load. A snubber may be required for the boost diode as well. The same empirical procedure applies. Snubbers were not necessary in this example.

Error Amplifier Configuration

R4, C18, C17

These components configure the error amplifier gain characteristics to accomplish a stable overall loop gain. One advantage of current mode control is the ability to close the loop with only three feedback components, R4, C18 and C17. The overall loop gain is the product of the modulator gain and the error amplifier gain. The DC modulator gain of the LM5118 is as follows:

$$DCGain_{(MOD)} = \frac{R_{LOAD} \times V_{IN}}{10R_S(V_{IN} + 2V_{OUT})}$$

The dominant, low frequency pole of the modulator is determined by the load resistance (R_{LOAD}) and output capacitance (C_{OUT}). The corner frequency of this pole is:

$$f_{P(MOD)} = \frac{1 + D_{MIN}}{2\pi \times R_{LOAD} \times C_{OUT}}$$

For this example, R_{LOAD} = 4Ω, D_{MIN} = 0.294, and C_{OUT} = 454 µF, therefore:

$f_{P(MOD)}$ = 149 Hz

DC $Gain_{(MOD)}$ =3.63 = 11.2 dB

Additionally, there is a right-half plane (RHP) zero associated with the modulator. The frequency of the RHP zero is:

$$f_{RHPzero} = \frac{R_{LOAD}\ (1 - D)^2}{2\pi \times L \times D}$$

$f_{RHPzero}$ = 7.8 kHz

The output capacitor ESR produces a zero given by:

$$ESR_{zero} = \frac{1}{2\pi \times ESR \times C_{OUT}}$$

ESR_{ZERO} = 70 kHz

The RHP zero complicates compensation. The best design approach is to reduce the loop gain to cross zero at about 30% of the calculated RHP zero frequency. The Type II error amplifier compensation provided by R4, C18 and C17 places one pole at the origin for high DC gain. The 2nd pole should

be located close to the RHP zero. The error amplifier zero (see below) should be placed near the dominate modulator pole. This is a good starting point for compensation. Refer to the on-line LM5118 Quick-Start calculator for ready to use equations and more details.

Components R4 and C18 configure the error amplifier as a type II configuration which has a DC pole and a zero at

$$f_z = \frac{1}{2 \times \pi \times R4 \times C18}$$

C17 introduces an additional pole used to cancel high frequency switching noise. The error amplifier zero cancels the modulator pole leaving a single pose response at the crossover frequency of the loop gain if the crossover frequency is much lower than the right half plane zero frequency. A single pole response at the crossover frequency yields a very stable loop with 90 degrees of phase margin.

For the design example, a target loop bandwidth (crossover frequency) of 2.0 kHz was selected (about 30% of the right-half-plane zero frequency). The error amplifier zero (fz) should be selected at a frequency near that of the modulator pole and much less than the target crossover frequency. This constrains the product of R4 and C18 for a desired compensation network zero to be less than 2 kHz. Increasing R4, while proportionally decreasing C18 increases the error amp gain. Conversely, decreasing R4 while proportionally increasing C18 decreases the error amp gain. For the design example C18 was selected for 4.7 nF and R4 was selected to be 10 kΩ. These values set the compensation network zero at 149 Hz. The overall loop gain can be predicted as the sum (in dB) of the modulator gain and the error amp gain.

If a network analyzer is available, the modulator gain can be measured and the error amplifier gain can be configured for the desired loop transfer function. If a network analyzer is not available, the error amplifier compensation components can be designed with the guidelines given. Step load transient tests can be performed to verify acceptable performance. The step load goal is minimal overshoot with a damped response.

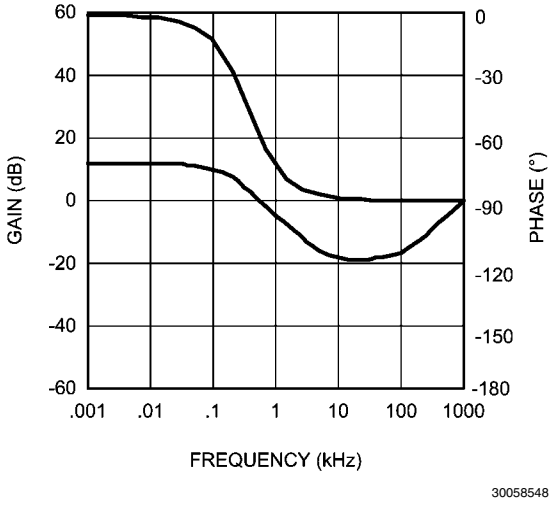


FIGURE 13. Modulator Gain and Phase

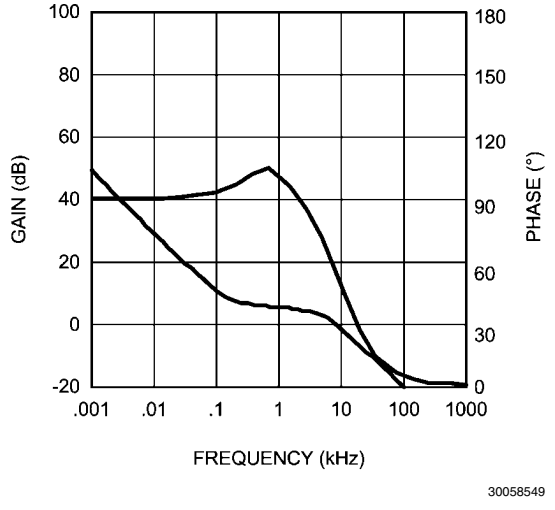


FIGURE 14. Error Amplifier Gain and Phase

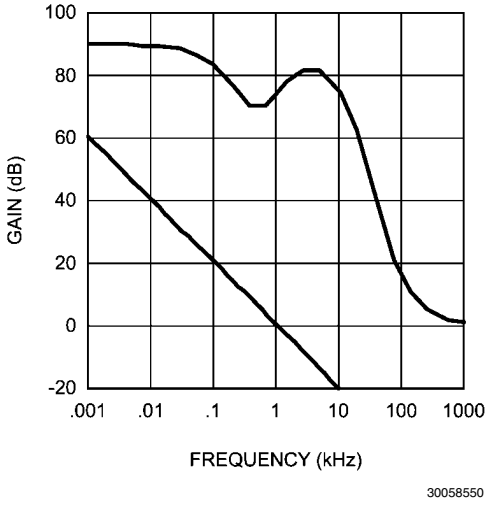


FIGURE 15. Overall Loop Gain and Phase

The plots shown in Figures 13, 14 and 15 illustrate the gain and phase diagrams of the design example. The overall bandwidth is lower in a buck-boost application due the compensation challenges associated with the right-half-plane zero. For a pure buck application, the bandwidth could be much higher. The LM5116 datasheet is a good reference for compensation design of a pure buck mode regulator.

Bias Power Dissipation Reduction

Buck or Buck-boost regulators operating with high input voltage can dissipate an appreciable amount of power while supplying the required bias current of the IC. The VCC regulator must step-down the input voltage VIN to a nominal VCC level of 7V. The large voltage drop across the VCC regulator translates into high power dissipation in the VCC regulator. There are several techniques that can significantly reduce this bias regulator power dissipation. Figures 16 and 17 depict two methods to bias the IC, one from the output voltage and one from a separate bias supply. In the first case, the internal VCC regulator is used to initially bias the VCC pin. After the output voltage is established, the VCC pin bias current is supplied through the VCCX pin, which effectively disables the internal VCC regulator. Any voltage greater than 4.0V can supply VCC bias through the VCCX pin. However, the voltage applied to the VCCX pin should never exceed 15V. The voltage supplied through VCCX must be large enough to drive the switching MOSFETs into full saturation.

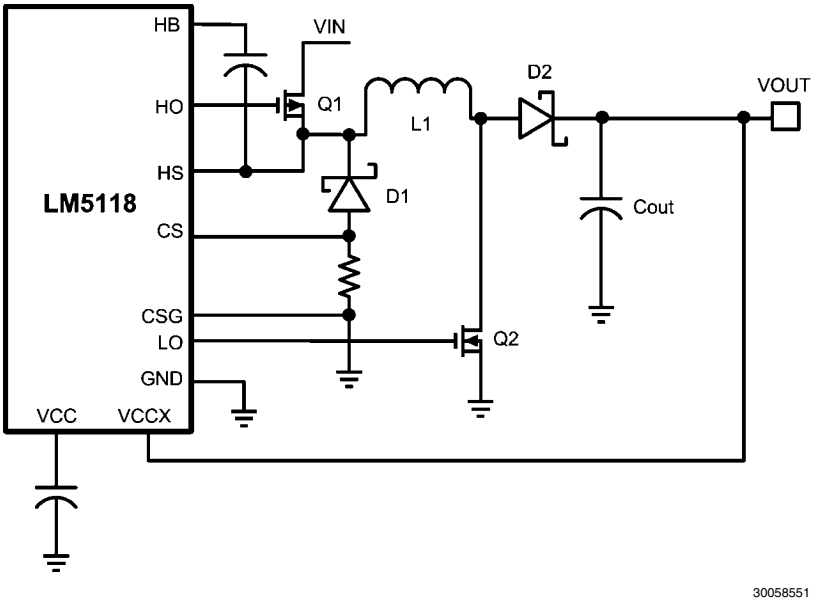


FIGURE 16. VCC Bias from VOUT 4V < VOUT < 15V

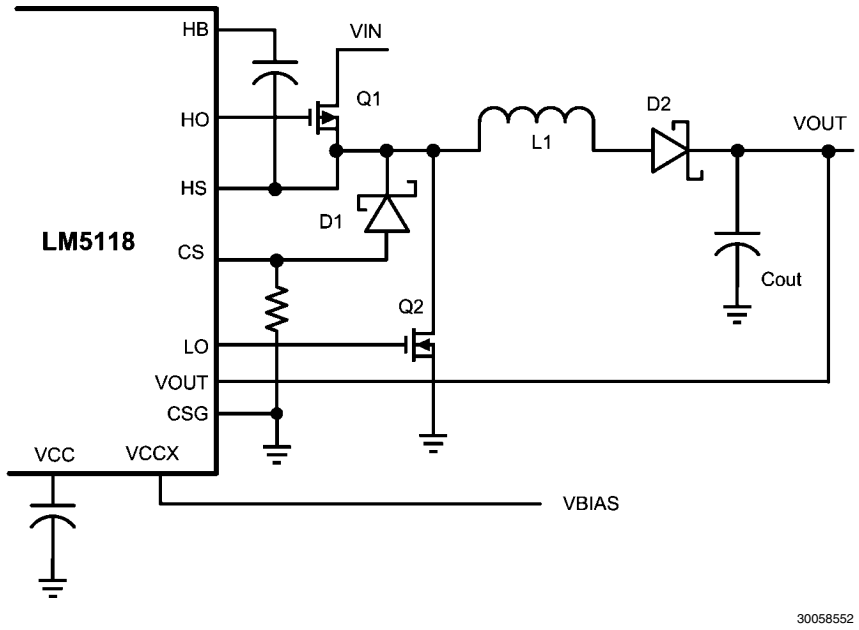


FIGURE 17. VCC Bias with Additional Bias Supply

PCB Layout and Thermal Considerations

In a buck-boost regulator, there are two loops where currents are switched very fast. The first loop starts from the input capacitors, and then to the buck switch, the inductor, the boost switch then back to the input capacitor. The second loop starts from the inductor, and then to the output diode, the output capacitor, the re-circulating diode, and back to the inductor. Minimizing the PC board area of these two loops reduces the stray inductance and minimizes noise and the possibility of erratic operation. A ground plane in the PC board is recommended as a means to connect the input filter capacitors to the output filter capacitors and the PGND pins of the LM5118. Connect all of the low current ground connections (C_{SS}, R_T, C_{RAMP}) directly to the regulator AGND pin. Connect the AGND and PGND pins together through topside copper area covering the entire underside of the device. Place several vias in this underside copper area to the ground plane of the input capacitors.

The highest power dissipating components are the two power MOSFETs, the re-circulating diode, and the output diode. The easiest way to determine the power dissipated in the MOSFETs is to measure the total conversion losses (P_{IN} - P_{OUT}), then subtract the power losses in the Schottky diodes, output inductor and any snubber resistors. An approximation for the re-circulating Schottky diode loss is:

$P = (1\text{-}D) \times I_{OUT} \times V_{FWD}$.

The boost diode loss is

$P = I_{OUT} \times V_{FWD}$.

If a snubber is used, the power loss can be estimated with an oscilloscope by observation of the resistor voltage drop at both turn-on and turn-off transitions. The LM5118 package has an exposed thermal pad to aid power dissipation. Selecting diodes with exposed pads will aid the power dissipation of the diodes as well. When selecting the MOSFETs, pay careful attention to $R_{DS(ON)}$ at high temperature. Also, selecting MOSFETs with low gate charge will result in lower switching losses.

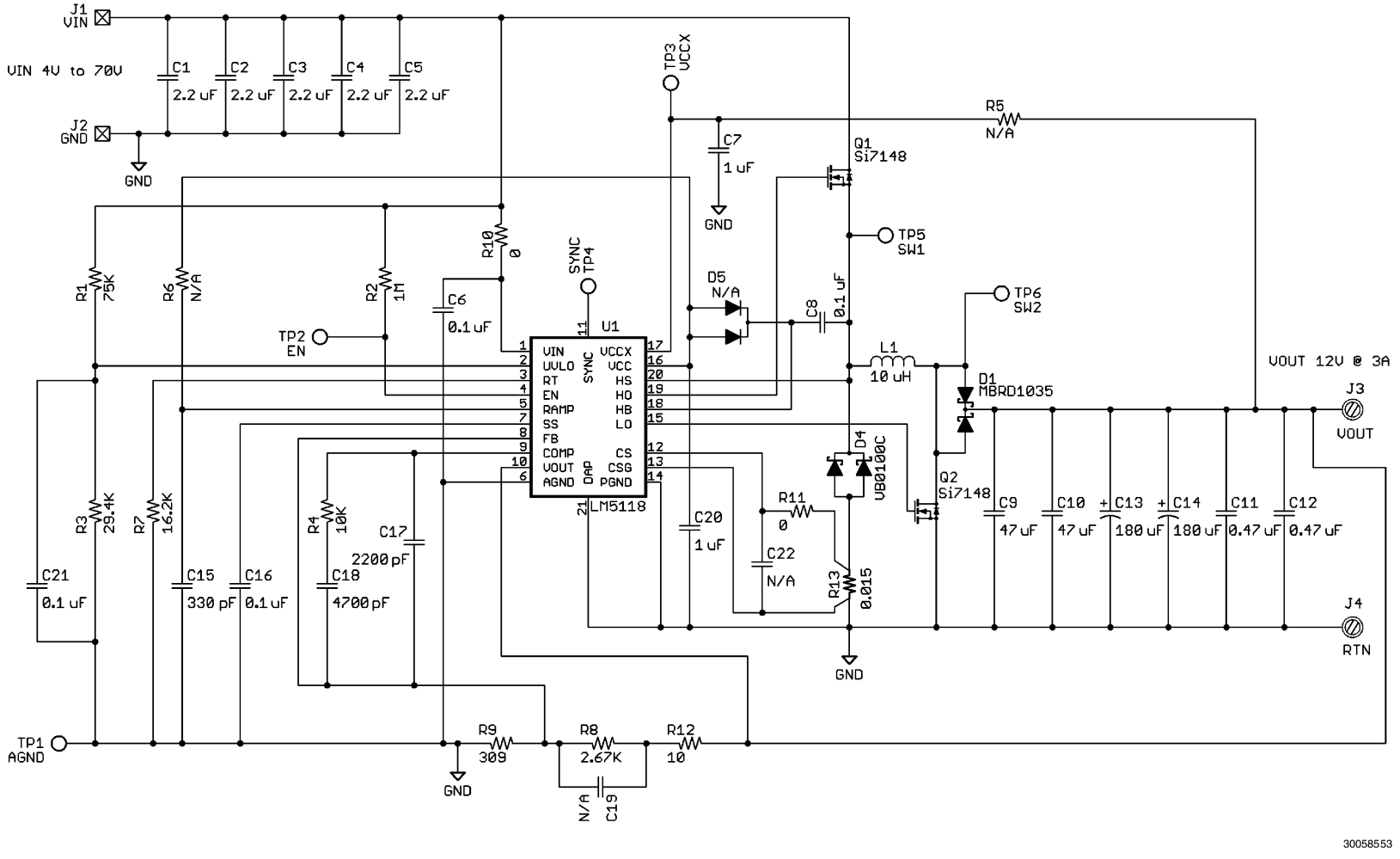
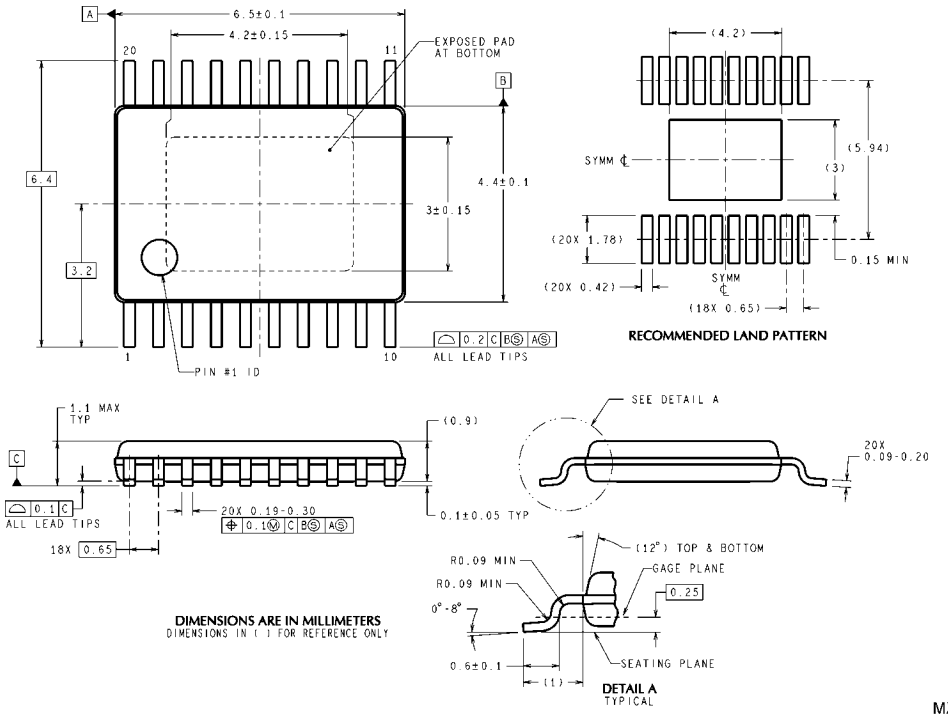


FIGURE 18. 12V, 3A Typical Application Schematic

Physical Dimensions inches (millimeters) unless otherwise noted



MXA20A (Rev C)

TSSOP-20EP Outline Drawing
NS Package Number MXA20A

Notes

Notes

For more National Semiconductor product information and proven design tools, visit the following Web sites at:

Products		Design Support	
Amplifiers	www.national.com/amplifiers	WEBENCH	www.national.com/webench
Audio	www.national.com/audio	Analog University	www.national.com/AU
Clock Conditioners	www.national.com/timing	App Notes	www.national.com/appnotes
Data Converters	www.national.com/adc	Distributors	www.national.com/contacts
Displays	www.national.com/displays	Green Compliance	www.national.com/quality/green
Ethernet	www.national.com/ethernet	Packaging	www.national.com/packaging
Interface	www.national.com/interface	Quality and Reliability	www.national.com/quality
LVDS	www.national.com/lvds	Reference Designs	www.national.com/refdesigns
Power Management	www.national.com/power	Feedback	www.national.com/feedback
Switching Regulators	www.national.com/switchers		
LDOs	www.national.com/ldo		
LED Lighting	www.national.com/led		
PowerWise	www.national.com/powerwise		
Serial Digital Interface (SDI)	www.national.com/sdi		
Temperature Sensors	www.national.com/tempsensors		
Wireless (PLL/VCO)	www.national.com/wireless		

THE CONTENTS OF THIS DOCUMENT ARE PROVIDED IN CONNECTION WITH NATIONAL SEMICONDUCTOR CORPORATION ("NATIONAL") PRODUCTS. NATIONAL MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS PUBLICATION AND RESERVES THE RIGHT TO MAKE CHANGES TO SPECIFICATIONS AND PRODUCT DESCRIPTIONS AT ANY TIME WITHOUT NOTICE. NO LICENSE, WHETHER EXPRESS, IMPLIED, ARISING BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT.

TESTING AND OTHER QUALITY CONTROLS ARE USED TO THE EXTENT NATIONAL DEEMS NECESSARY TO SUPPORT NATIONAL'S PRODUCT WARRANTY. EXCEPT WHERE MANDATED BY GOVERNMENT REQUIREMENTS, TESTING OF ALL PARAMETERS OF EACH PRODUCT IS NOT NECESSARILY PERFORMED. NATIONAL ASSUMES NO LIABILITY FOR APPLICATIONS ASSISTANCE OR BUYER PRODUCT DESIGN. BUYERS ARE RESPONSIBLE FOR THEIR PRODUCTS AND APPLICATIONS USING NATIONAL COMPONENTS. PRIOR TO USING OR DISTRIBUTING ANY PRODUCTS THAT INCLUDE NATIONAL COMPONENTS, BUYERS SHOULD PROVIDE ADEQUATE DESIGN, TESTING AND OPERATING SAFEGUARDS.

EXCEPT AS PROVIDED IN NATIONAL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, NATIONAL ASSUMES NO LIABILITY WHATSOEVER, AND NATIONAL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO THE SALE AND/OR USE OF NATIONAL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE CHIEF EXECUTIVE OFFICER AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

National Semiconductor and the National Semiconductor logo are registered trademarks of National Semiconductor Corporation. All other brand or product names may be trademarks or registered trademarks of their respective holders.

Copyright© 2008 National Semiconductor Corporation

For the most current product information visit us at www.national.com

National Semiconductor Americas Technical Support Center
Email: support@nsc.com
Tel: 1-800-272-9959

National Semiconductor Europe Technical Support Center
Email: europe.support@nsc.com
German Tel: +49 (0) 180 5010 771
English Tel: +44 (0) 870 850 4288

National Semiconductor Asia Pacific Technical Support Center
Email: ap.support@nsc.com

National Semiconductor Japan Technical Support Center
Email: jpn.feedback@nsc.com